

TESIS DOCTORAL



Universidad
Carlos III de Madrid

ESCUELA POLITÉCNICA SUPERIOR
DOCTORADO EN CIENCIA Y TECNOLOGÍA INFORMÁTICA

Medidas de productividad en los proyectos de desarrollo de software: una aproximación por puestos de trabajo

Autor:

D. Adrián Hernández López.

Directores:

Dr. D. Ángel García Crespo.

Dr. D. Ricardo Colomo Palacios.

Departamento de Informática.

Leganés, Febrero de 2014

TESIS DOCTORAL

Medidas de productividad en los proyectos de desarrollo de software: una aproximación por puestos de trabajo

Autor:

D. Adrián Hernández López.

Directores:

Dr. D. Ángel García Crespo.

Dr. D. Ricardo Colomo Palacios.

Firma del Tribunal Calificador:

	Nombre	Firma
Presidente:	Belén Ruiz Mezcuá	
Vocal:	Edmundo Tovar Caro	
Secretario:	Rafael Valencia García	
Calificación:		

En Leganés, de Febrero de 2014.

Resumen

La productividad es una medida, principalmente económica, creada a finales del siglo XVIII. Desde entonces, numerosas modificaciones se han realizado sobre la definición inicial y se han incorporando a diversas áreas de conocimiento. Dentro de la Ingeniería del Software (IS), la productividad comenzó a ser objeto de estudio a finales de los años 70, casi de forma paralela a la concepción de la misma y al inicio del estudio de conceptos relacionados, tales como la estimación de esfuerzo.

La medición de la productividad en IS ha sido ampliamente analizada a nivel de proyecto y organización, sin embargo a nivel de puesto de trabajo no ha sido tan investigada. En estos escasos estudios, las medidas utilizadas suelen ser las mismas medidas que las empleadas en niveles superiores de medición. En concreto, las medidas empleadas suelen ser ratios entre una medida de tamaño de producto (p. ej., líneas de código o puntos función) y una medida de esfuerzo o tiempo (p. ej., horas-hombre u horas). Este tipo de medidas son muy específicas y no reflejan la realidad del trabajo desempeñado en todo el proceso de desarrollo, ya que no tienen en cuenta las características inherentes a cada puesto de trabajo. Así pues, la eficacia de estas medidas, en este nivel de medición, parece estar en entredicho y la realización de estudios que aporten nuevas medidas de productividad en IS a nivel de puesto de trabajo cobra sentido.

En la presente tesis doctoral se ha analizado la situación actual de la medición de la productividad en IS a nivel de puesto de trabajo con el objetivo de crear nuevas medidas. Para conseguir este objetivo se ha realizado un estudio del estado de la cuestión utilizando una metodología clásica de revisión de referencias junto con una revisión sistemática de la literatura. Una vez analizado el estado de la cuestión se ha

planteado un conjunto de hipótesis relacionadas con la construcción de nuevas medidas de productividad:

Hipótesis 1. *En los puestos de trabajo involucrados en la ejecución de proyectos de desarrollo de software se emplean otras entradas, además del tiempo y el esfuerzo.*

Hipótesis 2. *Las entradas utilizadas son distintas para cada puesto de trabajo involucrado en la ejecución de proyectos de desarrollo de software.*

Hipótesis 3. *En los puestos de trabajo involucrados en la ejecución de proyectos de desarrollo de software se producen otras salidas, además de líneas de código y funcionalidad.*

Hipótesis 4. *Las salidas producidas son distintas para cada puesto de trabajo involucrado en la ejecución de proyectos de desarrollo de software.*

Hipótesis 5. *Las medidas de productividad más utilizadas a nivel de puesto de trabajo en los proyectos de desarrollo de software tienen una eficacia limitada para medir la productividad real de los trabajadores.*

Hipótesis 6. *Es posible medir de forma más eficaz la productividad de los puestos de trabajo en los proyectos de desarrollo de software con nuevas medidas que combinen varios elementos: entradas, salidas y factores.*

Tras el análisis del estado de la cuestión, se ha realizado una fase de investigación cualitativa mediante el empleo de entrevistas a trabajadores de IS y un posterior análisis de contenido, con el fin de obtener información suficiente para: (1) contrastar las cuatro primeras hipótesis con información cualitativa, y (2) construir el medio de

recogida de información para la siguiente fase de la investigación. Con respecto al primer objetivo, ha sido posible contrastar dos hipótesis (H1 y H3). En la segunda fase, mediante una metodología cuantitativa, se han contrastado las cuatro primeras hipótesis planteadas. Para la recogida de información se ha utilizado un formulario construido a partir de los resultados de la fase cualitativa. Los resultados de esta fase indican que en los puestos de trabajo analizados (programador, analista, consultor, y jefe de proyecto): se utilizan otros recursos además del tiempo, se producen otras salidas además del código fuente y la funcionalidad entregada al cliente. Además, se han encontrado diferencias en el grado de uso de las entradas y en la producción de las salidas, por lo que el uso de una misma medida de productividad para todos los puestos bajo estudio es, en principio, ilógico.

Para contrastar las dos, y últimas, hipótesis se han construido nuevas medidas de productividad, teniendo en cuenta los resultados previos. En concreto, se ha utilizado *Data Envelopment Analysis* (DEA) como metodología personalizable para medir la productividad; y se han realizado cuatro casos de estudio empleando dicha metodología. Los resultados tras los casos de estudio indican que mediante DEA es posible medir la productividad de los puestos de trabajo vinculados con los proyectos de desarrollo y mantenimiento de software de forma más eficaz que con las medidas más utilizadas. Además, esta metodología permite conocer los puntos de mejora para que los trabajadores menos productivos aumenten su productividad, lo que supone una gran ventaja frente a otras medidas de productividad si el objetivo de medir, como es lógico suponer, es mejorar la productividad, y no simplemente evaluarla. Así pues, se contrastan las dos últimas hipótesis y se insta, entre otras futuras líneas de investigación, a continuar con nuevos estudios que comparen el uso de DEA con otras medidas de productividad.

Finalmente, se concluye que la medición de la productividad en los puestos de trabajo vinculados con los proyectos de desarrollo y mantenimiento de software continua siendo un reto. Para reducir la dificultad de éste, la presente tesis doctoral arroja luz aportando un marco de trabajo para analizar y plantear nuevas medidas de productividad, tanto en estos puestos de trabajo como en otros.

Abstract

Productivity is mainly an economic measure, created in the late eighteenth century. Since then, many changes have been made on its initial definition and have been incorporated into various areas of knowledge. Within Software Engineering (SE), productivity began to be studied in the late '70s. These efforts ran parallel to SE developments, such as effort estimation.

Measuring productivity in SE has been extensively analyzed at the project and organization level; however job level has not been investigated with the same depth. In these few studies, the measures used are often the same ones than those used in higher levels of measurement. Specifically, the measures employed are usually ratios between a measure of product size (e.g., lines of code or function points) and a measure of effort or time (e.g., man-hours or hours). Such measures do not reflect the reality of the work performed throughout the development process because they do not take into account the inherent characteristics of each job. Thus, the effectiveness of these measures, in this measurement level, seems to be in question and studies that provide new measures of productivity at job level make sense.

In this thesis we have analyzed the current state of productivity measurement at job level within SE with the goal of creating new measures. In order to achieve this objective a study of the state of the art has been carried out with a classical methodology along with a systematic review of the literature. After analyzing the state of the art, a number of hypotheses related to the construction of new productivity measures have been stated:

Hypothesis 1. In the jobs involved in the implementation of software development projects other inputs are used in addition to time and effort.

Hypothesis 2. The inputs used are different for every job involved in software development projects.

Hypothesis 3. In the jobs involved in the implementation of software development projects other outputs are produced in addition to source code lines and functionality.

Hypothesis 4. The outputs produced are different for every job involved in software development projects.

Hypothesis 5. The most used productivity measures at job level in software development projects have limited effectiveness for measuring real productivity of workers.

Hypothesis 6. It is possible to measure more effectively the productivity of jobs in software development projects with new measures that combine several elements: inputs, outputs and factors.

After analyzing the state of the art, a qualitative phase has been performed using interviews with SE workers and a subsequent content analysis of them in order to obtain pertinent information: (1) to test the first four hypotheses with qualitative information, and (2) to build the information gathering instrument for the next phase of research. Regarding the first objective, it has been possible to test two hypotheses (H1 and H3). In the second phase, using a quantitative method, the first four hypotheses have been contrasted and accepted. For the information gathering a form constructed from the results of the qualitative phase has been used. The results of this phase indicate that the analyzed job positions (programmer, analyst, consultant, and project manager): use other resources in addition to time, and deliver other outputs in addition to source code and functionality delivered to the client. Also some differences in the degree of use of inputs and production of outputs have been found. Therefore, the use of the same measure of productivity for all positions under study is, in principle, illogical.

To contrast the last two hypotheses new productivity measures have been built taking into account the previous results. Specifically, a customizable methodology for measuring productivity such as Data Envelopment Analysis (DEA) was used in four case studies. The results after these studies indicate that using DEA is a mean to measure the productivity of job level for job positions related to the development and maintenance of software projects in a more effectively way. Furthermore, this methodology allows knowing the points for improvement for the least productive workers in order to increase their productivity. This knowledge is a great advantage over other productivity measures if the goal of measuring, as is logical to assume, is to improve productivity, not simply to evaluate it. So the last two hypotheses has been supported. Consequently we call, among other future research, to continue with further studies comparing the use of DEA with other measures of productivity.

Finally, it is concluded that the measurement of productivity in job positions related with software development and maintenance projects remains a challenge. To reduce this difficulty, this thesis sheds some light on the topic by providing a framework to analyze and propose new measures of productivity for SE job roles.

Agradecimientos

Quiero dar las gracias a todas las personas que han apoyado, en mayor o menor medida, la realización de esta tesis doctoral. Con su apoyo y esfuerzo ha sido posible llegar hasta el final y cumplir con los objetivos inicialmente marcados. En especial doy las gracias a mi familia por su apoyo constante, tanto anímico, como económico ante la ausencia de becas para el desarrollo de la investigación; a todos los participantes en las investigaciones realizadas a lo largo de esta tesis, ya que sin ellos no hubiera sido posible obtener datos; y a las empresas que han participado en la fase de validación, que han aportado esfuerzo y voluntad, sin obtener una contrapartida por falta de financiación a la investigación.

Índice

RESUMEN.....	VI
ABSTRACT.....	IX
AGRADECIMIENTOS	XIII
ÍNDICE.....	XV
ÍNDICE DE FIGURAS.....	XXIII
ÍNDICE DE TABLAS.....	XXV
1 INTRODUCCIÓN	3
1.1. Contexto.....	3
1.2. Definición del problema y motivación	7
1.3. Objetivos de investigación	11
1.4. Hipótesis de investigación.....	12
1.5. Aproximación a la solución.....	15
1.6. Método de investigación	16
1.7. Aportaciones de la investigación	20
1.8. Validez de la solución	22
1.9. Estructura de la tesis doctoral.....	24

2	REVISIÓN DEL ESTADO DE LA CUESTIÓN	29
2.1.	Introducción	29
2.2.	Elementos necesarios para la medición de la productividad	36
2.2.1.	Definición de productividad.....	39
2.2.2.	Factores.....	41
2.2.3.	Entradas y salidas.....	50
2.2.3.1.	Motivación y satisfacción laboral.....	52
2.3.	Cuestiones	58
2.4.	Puestos de trabajo.....	62
2.4.1.	Puestos de trabajo en proyectos de desarrollo de software	65
2.4.2.	Puestos de trabajo según Métrica 3.....	69
2.5.	Medición de la productividad a nivel de puesto de trabajo en proyectos de desarrollo de software	73
2.6.	Revisión sistemática de la literatura.....	76
2.6.1.	Preguntas de investigación.....	78
2.6.2.	Fuentes de consulta.....	78
2.6.3.	Cadena de búsqueda y delimitaciones temporales.....	79
2.6.4.	Proceso de recuperación, filtrado y revisión	80
2.6.5.	Ejecución	81
2.6.5.1.	Estudios incluidos y excluidos - Fase 1.....	82
2.6.5.2.	Estudios incluidos y excluidos - Fase 2.....	83
2.6.6.	Resultados	84

2.6.7.	Hallazgos.....	85
2.6.8.	Breve discusión de la revisión sistemática	89
2.6.9.	Conclusiones	91
2.7.	Resumen del estado de la cuestión	94
3	ENFOQUE DEL PROBLEMA	101
3.1.	Introducción	101
3.2.	Justificación y método	102
3.3.	Fase cualitativa	103
3.3.1.	Metodología	104
3.3.2.	Resultados	106
3.3.2.1.	Definición de productividad	106
3.3.2.2.	Entradas y salidas	117
3.3.2.3.	Satisfacción laboral y motivación	124
3.3.2.4.	Factores que afectan a la productividad	137
3.3.2.5.	Conocimiento de las medidas convencionales por parte de los trabajadores.....	140
3.3.3.	Contraste de hipótesis	141
3.3.4.	Validez y fiabilidad.....	142
3.3.5.	Conclusiones	143
3.4.	Fase cuantitativa.....	148
3.4.1.	Metodología	149

3.4.2.	Recogida muestral.....	151
3.4.3.	Resultados.....	152
3.4.3.1.	Descriptivos.....	152
3.4.4.	Contraste de hipótesis.....	158
3.4.5.	Situación tras analizar los resultados.....	166
3.4.6.	Validez y fiabilidad	169
3.5.	Situación tras ambas fases de estudio.....	170
4	SOLUCIÓN PROPUESTA	177
4.1.	Introducción	177
4.2.	Data Envelopment Analysis (DEA).....	179
4.2.1.	Características de los modelos DEA.....	182
4.2.1.1.	Ventajas y desventajas de DEA.....	183
4.2.2.	DEA en IS	185
4.2.3.	DEA para medir la productividad de los trabajadores del conocimiento	186
4.3.	Solución propuesta basada en DEA	187
5	VALIDACIÓN	197
5.1.	Introducción	197
5.2.	Casos de estudio.....	199
5.2.1.	Caso 1 - Prueba piloto	200

5.2.1.1.	Descripción de organización y proceso de trabajo.....	200
5.2.1.2.	Puestos de trabajo, variables y factores a analizar	200
5.2.1.3.	Descripción demográfica de participantes	201
5.2.1.4.	Preparación de datos	202
5.2.1.5.	Generación y análisis de resultados.....	203
5.2.1.6.	Conclusiones parciales	203
5.2.2.	Caso de estudio 2	204
5.2.2.1.	Descripción de organización y proceso de trabajo.....	204
5.2.2.2.	Puestos de trabajo, variables y factores a analizar	205
5.2.2.3.	Descripción demográfica de participantes	210
5.2.2.4.	Preparación de datos	211
5.2.2.5.	Generación y análisis de resultados.....	215
5.2.2.6.	Conclusiones parciales	223
5.2.3.	Caso de estudio 3	224
5.2.3.1.	Descripción de organización y proceso de trabajo.....	224
5.2.3.2.	Puestos de trabajo, variables y factores a analizar	225
5.2.3.3.	Descripción demográfica de participantes	229
5.2.3.4.	Preparación de datos	230
5.2.3.5.	Generación y análisis de resultados.....	230
5.2.3.6.	Conclusiones parciales	230
5.2.4.	Caso de estudio 4	230

5.2.4.1.	Descripción de organización y proceso de trabajo	230
5.2.4.2.	Puestos de trabajo, variables y factores a analizar.....	231
5.2.4.3.	Descripción demográfica de participantes.....	235
5.2.4.4.	Preparación de datos.....	236
5.2.4.5.	Generación y análisis de resultados	237
5.2.4.6.	Conclusiones parciales	238
5.3.	Conclusiones globales a los casos de estudio.....	239
5.3.1.	Limitaciones	239
5.3.2.	Ventajas y desventajas.....	240
5.3.3.	Recomendaciones.....	241
5.3.4.	Contraste de hipótesis de investigación.....	242
6	CONCLUSIONES Y LÍNEAS FUTURAS	247
6.1.	Conclusiones	247
6.2.	Líneas futuras	249
	BIBLIOGRAFÍA	255
	ANEXOS	283
	Anexo A - Protocolo revisión sistemática.....	287
	Anexo B - Listas de resultados de la revisión sistemática	301

Anexo C - Guión de entrevista semiestructurada.....	347
Anexo D - Cuestionario y justificación de selección de elementos	353
Anexo E - Condiciones para aplicar ANOVA	365
Anexo F - Publicaciones relacionadas con la tesis	373
Anexo G - Diagramas de procesos de proyectos de desarrollo software en las organizaciones participantes	379
Anexo H - Cuestionarios utilizados en los casos de estudio	385

Índice de figuras

ILUSTRACIÓN 1. EL MODELO DE CAPAS DE COMPORTAMIENTO DE DESARROLLO DE SOFTWARE (KLING & SCACCHI, 1982).....	5
ILUSTRACIÓN 2. APROXIMACIÓN A LA SOLUCIÓN	17
ILUSTRACIÓN 3. APROXIMACIÓN A LA SOLUCIÓN - ETAPA 1 (REVISIÓN DEL ESTADO DE LA CUESTIÓN)	18
ILUSTRACIÓN 4. APROXIMACIÓN A LA SOLUCIÓN - ETAPA 2 (ENFOQUE DEL PROBLEMA).....	19
ILUSTRACIÓN 5. APROXIMACIÓN A LA SOLUCIÓN - ETAPA 3 (DESARROLLO Y VALIDACIÓN DE NUEVAS MEDIDAS)	20
ILUSTRACIÓN 6. PIRÁMIDE DE NECESIDADES DE MASLOW (1943).....	54
ILUSTRACIÓN 7. TEORÍA DUAL DE HERZBERG (1959).....	55
ILUSTRACIÓN 8. PROCESO DE FILTRADO; BASADO EN (AFZAL, ET AL., 2009).....	81
ILUSTRACIÓN 9. PROCESO DE ENFOQUE DEL PROBLEMA	103
ILUSTRACIÓN 10. PROCESO DE TRABAJO DE LA FASE CUALITATIVA	105
ILUSTRACIÓN 11. TÉRMINOS PARA LA DEFINICIÓN DE PRODUCTIVIDAD.....	108
ILUSTRACIÓN 12. SATISFACCIÓN CON EL TIPO DE TRABAJO	127
ILUSTRACIÓN 13. SATISFACCIÓN CON JEFES Y SUPERIORES.....	127
ILUSTRACIÓN 14. SATISFACCIÓN CON LA ORGANIZACIÓN DEL TRABAJO	127
ILUSTRACIÓN 15. MEJORA DE LA SATISFACCIÓN LABORAL.....	131
ILUSTRACIÓN 16. MEJORA DE LA MOTIVACIÓN	133
ILUSTRACIÓN 17. PRINCIPALES ELEMENTOS DE LA SATISFACCIÓN LABORAL.....	134
ILUSTRACIÓN 18. PROCESO FASE CUANTITATIVA.....	149
ILUSTRACIÓN 19. SOLUCIÓN PROPUESTA BASADA EN DEA	188
ILUSTRACIÓN 20. ORGANIGRAMA DE PUESTOS EN EL CASO 2.....	206
ILUSTRACIÓN 21. NÚMERO DE DMUS INEFICIENTES QUE CONSIDERAN A OTRA DMU COMO MODELO DE REFERENCIA	217

ILUSTRACIÓN 22. NÚMERO DE DMUS INEFICIENTES QUE CONSIDERAN A OTRA DMU COMO MODELO DE REFERENCIA (SBM)	218
ILUSTRACIÓN 23. NÚMERO DE TRABAJADORES INEFICIENTES QUE CONSIDERAN A OTRO COMO MODELO DE REFERENCIA.....	220
ILUSTRACIÓN 24. NÚMERO DE DMUS INEFICIENTES QUE CONSIDERAN A OTRA DMU COMO MODELO DE REFERENCIA (SBM)	222
ILUSTRACIÓN 25. ORGANIGRAMA DE PUESTOS EN EL CASO 3	227
ILUSTRACIÓN 26. PROCESO DE FILTRADO; BASADO EN (AFZAL, TORKAR & FELDT, 2009)	294
ILUSTRACIÓN 27. PROCESO DE TRABAJO EN LOS PROYECTOS DE DESARROLLO Y MANTENIMIENTO DEL CASO 2.....	380
ILUSTRACIÓN 28. PROCESO DE TRABAJO EN LOS PROYECTOS DE DESARROLLO Y MANTENIMIENTO DEL CASO 3.....	381
ILUSTRACIÓN 29. PROCESO DE TRABAJO EN LOS PROYECTOS DE DESARROLLO Y MANTENIMIENTO DEL CASO 4.....	382

Índice de tablas

TABLA 1. PROBLEMAS IDENTIFICADOS - SOLUCIÓN PROPUESTA.....	10
TABLA 2. HIPÓTESIS Y PROBLEMAS DE INVESTIGACIÓN RELACIONADOS.....	14
TABLA 3. FACTORES INCIDENTES EN LA PRODUCTIVIDAD.....	48
TABLA 4. ENTRADAS Y SALIDAS UTILIZADAS PARA MEDIR LA PRODUCTIVIDAD	61
TABLA 5. RESULTADOS FASE 1 – SIN FILTRAR (16/05/2011)	82
TABLA 6. RESULTADOS FASE 1 – PRIMER FILTRO (16/05/2011).....	83
TABLA 7. RESULTADOS TRAS EJECUTAR LA FASE 2.....	84
TABLA 8. RESULTADOS CUANTITATIVOS	84
TABLA 9. TÉRMINOS DE LA DEFINICIÓN DE PRODUCTIVIDAD PARA INGENIERO DE SOFTWARE.	109
TABLA 10. TÉRMINOS DE LA DEFINICIÓN DE PRODUCTIVIDAD PARA JEFE DE PROYECTO.....	110
TABLA 11. TÉRMINOS DE LA DEFINICIÓN DE PRODUCTIVIDAD PARA PROYECTO	110
TABLA 12. TÉRMINOS DE LA DEFINICIÓN DE PRODUCTIVIDAD PARA ORGANIZACIÓN.....	111
TABLA 13. ENTRADAS MENCIONADAS POR LOS PARTICIPANTES.....	120
TABLA 14. SALIDAS MENCIONADAS POR LOS PARTICIPANTES.....	122
TABLA 15. SATISFACCIÓN LABORAL	125
TABLA 16. FACTORES RELACIONADOS CON LA SATISFACCIÓN LABORAL.....	127
TABLA 17. ELEMENTOS QUE MEJORAN LA SATISFACCIÓN LABORAL	131
TABLA 18. MOTIVADORES.....	132
TABLA 19. DESMOTIVADORES.....	133
TABLA 20. ELEMENTOS QUE MEJORAN LA PRODUCTIVIDAD	139
TABLA 21. ESTADO DE LAS HIPÓTESIS TRAS EJECUTAR LA FASE CUALITATIVA.....	148
TABLA 22. ESTADÍSTICOS DESCRIPTIVOS DE LAS ENTRADAS	154
TABLA 23. ESTADÍSTICOS DESCRIPTIVOS DE LAS SALIDAS.....	155
TABLA 24. DESVIACIÓN SEMI-INTERCUARTÍLICA Y MEDIANA DE LAS ENTRADAS.....	156
TABLA 25. DESVIACIÓN SEMI-INTERCUARTÍLICA Y MEDIANA DE LAS SALIDAS.....	157

TABLA 26. RESULTADOS DEL TEST DE KRUSKAL WALLIS (VARIABLE DE AGRUPACIÓN: PUESTO DE TRABAJO)	160
TABLA 27. DIFERENCIAS TEÓRICAS (PRUEBA DE DUNN)	162
TABLA 28. RANGOS MEDIOS Y DIFERENCIAS OBSERVADAS PARA LAS ENTRADAS (KRUSKAL WALLIS, CON $P < 0.05$)	164
TABLA 29. RANGOS MEDIOS Y DIFERENCIAS OBSERVADAS PARA LAS SALIDAS (CON $P < 0.05$ EN KRUSKAL WALLIS)	166
TABLA 30. ESTADO DE LAS HIPÓTESIS TRAS EJECUTAR LA FASE CUANTITATIVA.....	173
TABLA 31. VARIABLES BAJO ESTUDIO EN EL CASO 1	201
TABLA 32. DESCRIPCIÓN DEMOGRÁFICA DE LOS PARTICIPANTES EN EL CASO 1	202
TABLA 33. PARTICIPANTES EN EL CASO 1	202
TABLA 34. COEFICIENTES DE CORRELACIÓN (CASO 1).....	202
TABLA 35. VARIABLES BAJO ESTUDIO EN EL CASO 2.....	208
TABLA 36. DESCRIPCIÓN DEMOGRÁFICA DE LOS PARTICIPANTES (INICIALES) EN EL CASO 2.....	210
TABLA 37. DESCRIPCIÓN DEMOGRÁFICA DE LOS PARTICIPANTES (FINALES) EN EL CASO 2	211
TABLA 38. CORRELACIONES ENTRE VARIABLES EN EL CASO DE ESTUDIO 2 (1/2).....	212
TABLA 39. CORRELACIONES ENTRE VARIABLES EN EL CASO DE ESTUDIO 2 (2/2).....	213
TABLA 39. CORRELACIONES (A PARTIR DE VALORES SIN TRANSFORMAR).....	215
TABLA 40. VALORES PARA LAS DMUS (CASO 2).....	215
TABLA 41. PRODUCTIVIDAD DE DESARROLLO DE SOFTWARE (BCC)	217
TABLA 42. PRODUCTIVIDAD DE DESARROLLO DE SOFTWARE (SBM)	218
TABLA 43. HOLGURAS MEDIANTE SBM EN PRODUCTIVIDAD DE DESARROLLO	219
TABLA 44. PRODUCTIVIDAD DE ANÁLISIS DE NECESIDADES (BCC).....	220
TABLA 45. PRODUCTIVIDAD DE DESARROLLO DE ANÁLISIS DE NECESIDADES (SBM)	221
TABLA 46. HOLGURAS MEDIANTE SBM EN PRODUCTIVIDAD DE ANÁLISIS DE NECESIDADES.....	222
TABLA 47. VARIABLES BAJO ESTUDIO EN EL CASO 3.....	228
TABLA 48. VARIABLES BAJO ESTUDIO EN EL CASO 4.....	234
TABLA 49. DESCRIPCIÓN DEMOGRÁFICA DE LOS PARTICIPANTES EN EL CASO 4	235
TABLA 50. CORRELACIONES ENTRE VARIABLES CASO 4	237
TABLA 51. RESULTADOS DE PRUEBAS PRE-REQUISITO PARA APLICAR PRUEBAS PARAMÉTRICAS	367
TABLA 52. PRUEBA DE KOLMOGOROV-SMIRNOV (NORMALIDAD).....	369

TABLA 53. PRUEBA DE LEVENE (HOMOCEDASTICIDAD)	370
---	-----

1 Introducción

1.1. Contexto

Los orígenes del término “productividad” se remontan al siglo XVIII. Este término fue introducido por Quesnay (1766). Sin embargo, hasta mediados del siglo pasado, sus definiciones eran confusas. Tradicionalmente, la productividad ha sido definida como el ratio de salidas producidas por unidad de entrada (Jefferys, Hausberger, & Lindblad, 1954). Junto con la calidad y la rentabilidad, la productividad, forman una triplete en la que todas las partes están relacionadas con el mismo fenómeno: el resultado económico de la organización (Gummesson, 1995). Esta definición, basada en un ratio entre las salidas y las entradas, encaja perfectamente en el paradigma de la fabricación dado que este tipo de producciones están basadas en cantidades de unidades de medida estandarizadas y claramente identificadas, tanto para las entradas como las salidas. Sin embargo, no encaja con tanta perfección en las industrias de servicios, ni en la industria Tecnologías de la Información y Comunicación (TIC). En estas industrias, es difícil medir la productividad de los puestos de trabajo, dado que las tareas que en estos se desempeñan no están estandarizadas, no tienen tiempos estándar de producción, pueden ser desempeñadas por varios trabajadores, no son fácilmente observables, y generan y utilizan elementos tangibles e intangibles (Davenport & Prusak, 2000). Por ello, la gestión de la productividad es un reto dentro

de este tipo de industrias (Drucker, 1999). Este reto está presente de igual forma en los proyectos TIC. Mientras que en las industria manufactureras se han diseñado y probado métodos para determinar la productividad, el sector TIC está un paso por detrás en términos de disponer de métodos para evaluar las salidas y predecir el esfuerzo necesario para completar los proyectos (Dalcher, 2006). Además, el sector TIC puede considerarse en parte como un sector de servicios, y como Grönroos y Ojasalo (2004) concluyen, la noción de la productividad en las industrias de servicios tiene el efecto combinado de la efectividad del proveedor de servicios, que gestiona la rentabilidad de sus recursos y procesos de producción (eficiencia interna), y la calidad percibida de sus servicios (eficiencia externa), lo que hacen de la productividad un concepto diferente en comparación con el concepto tradicional de productividad en la industria de fabricación.

Dentro del sector TIC, se encuentra la Ingeniería del Software (IS), cuya misión es la aplicación de un enfoque sistemático, disciplinado y cuantificable al desarrollo, operación y mantenimiento de software, y el estudio de estos enfoques (Society, 2004). En IS, la investigación sobre medidas de productividad se ha centrado principalmente en la medición a nivel de proyecto (Anselmo & Ledgard, 2003; Petersen, 2011), y existe cierta carencia de investigación sobre medidas de productividad a niveles inferiores, por ejemplo a nivel de equipos de trabajo, o a nivel de puesto de trabajo (Pfleeger, 2008). A modo de ejemplo de estas carencias, se puede destacar como la importancia del diseño en los procesos de producción de software ha sido estudiada (ver p. ej., Anselmo & Ledgard, 2003), pero medir la productividad de los diseñadores de software, e incluso de las tareas de diseño de software, sigue siendo un reto desde hace años (Lowry, 2010; Rombach, 1990). Este ejemplo puede ser extrapolado a otras áreas de IS donde las salidas producidas continúan sin ser medidas ni valoradas en el contexto de la productividad. Los niveles de medición en IS pueden comprenderse mejor utilizando el modelo de comportamiento en capas de desarrollo de software (*layered behavioral model of software development*) (Kling & Scacchi, 1982) en el que se incluyen las siguientes capas (niveles): individual, equipo, proyecto, compañía,

entorno del negocio (ver Ilustración 1). Se denomina modelo de capas de comportamiento a esta aproximación dado que el desarrollo de software está generado por humanos en lugar de por máquinas, por lo que su creación debe ser analizada como un proceso de comportamiento (Curtis, Krasner, & Iscoe, 1988).

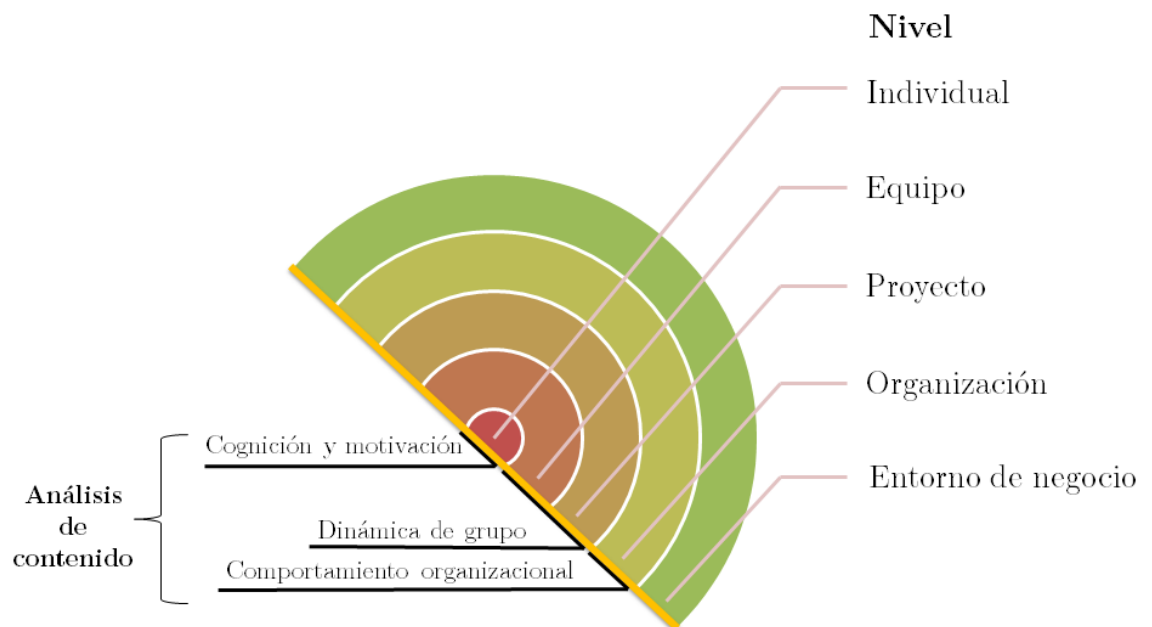


Ilustración 1. El modelo de capas de comportamiento de desarrollo de software (Kling & Scacchi, 1982)

La medición de la productividad en IS está estrechamente vinculada con la estimación del esfuerzo requerido para la ejecución de los proyectos y su dimensionamiento (Fenton & Neil, 1999). Paralelamente a la evolución de las medidas de dimensión de proyectos en IS, desde las SLOC (*Source Lines of Code*) que miden el tamaño del proyecto basándose en las líneas de código que lo componen, a los Puntos Función (PF) en alguna de sus variantes que miden la cantidad de funcionalidad incluida en el proyecto, las medidas de productividad fueron evolucionando desde SLOC/horas-hombre a PF/horas-hombre. Dado que la estimación en IS continúa siendo un reto, el estudio de nuevas medidas de productividad abre alternativas en dicho área. Además, la mejora de la productividad en el desarrollo de software no puede ser realizada sin una medición de la productividad (Anselmo & Ledgard, 2003).

Desde el punto de vista de los puestos de trabajo, incluidos y relacionados con IS, es posible decir que éstos necesitan una alta carga de capital humano para su desempeño (Acuña, Juristo, & Moreno, 2006; Colomo-Palacios, Tovar-Caro, García-Crespo, & Gómez-Berbís, 2010). Esto se debe a que, tal y como ya se ha mencionado, el desarrollo de software está generado por humanos en lugar de por máquinas (Curtis, et al., 1988). La importancia de este capital en el desarrollo de software ha sido reflejada en las actividades relacionadas con la productividad tales como los modelos de estimación de desarrollo de software. Por ejemplo, Boehm incluyó en su famoso modelo *Constructive Cost Model* (COCOMO) algunos factores (*drivers*) relacionados con el capital humano como la continuidad del personal, la cohesión del equipo, o la experiencia (Boehm, 1981b; Boehm et al., 2000).

Dentro de la gestión de capital humano, la medición de la productividad representa un reto desde hace años que no ha llegado a ser cubierto en su totalidad (Drucker, 1999). Algunos investigadores han analizado el estado de la cuestión de forma general, por ejemplo la productividad de los trabajadores del conocimiento (Erne, 2011; Nachum, 1999; Ramirez & Nembhard, 2004), pero en IS existe una carencia de investigación con respecto a este objetivo. No obstante, la investigación a niveles superiores de medición continúa siendo un área de gran interés (Petersen, 2011). En los estudios sobre medición de productividad de carácter más general (ver p. ej.: Ramirez & Nembhard, 2004), se menciona la medición de la productividad en IS mediante la utilización de los PF (salida) y una medida de esfuerzo (entrada). Así pues, parece lógico y necesario pensar en la continuidad de la investigación en el área de la medición de la productividad en IS.

En este escenario, la definición de nuevos modelos, medidas y métodos de evaluación pueden dilucidar la productividad en los proyectos IS, especialmente en los proyectos de desarrollo de software (Petersen, 2011). A su vez, las organizaciones necesitan conocer sus resultados de productividad, junto con los resultados del sector, para realizar un *benchmarking* (evaluación comparativa) de sus situación (Maxwell &

Forselius, 2000). Además, es necesario contemplar factores *soft*, es decir, factores relacionados con el capital intelectual en el campo de la medición en IS (Pfleeger, 2008).

Por todo lo anterior, y con el objetivo de establecer nuevas medidas de productividad a nivel de puesto de trabajo en los proyectos de desarrollo de software, se define, a continuación, el problema en cuestión y la motivación para afrontarlo, los objetivos de la investigación, la aproximación a la solución y los métodos utilizados para alcanzarla, y finalmente las aportaciones y validez de la solución.

1.2. Definición del problema y motivación

El problema de investigación se identifica mediante el análisis de las dificultades y necesidades encontradas en la literatura reciente, en el contexto de la medición de la productividad a nivel de puesto de trabajo dentro los proyectos de desarrollo de software. Tal y como apuntaba Boehm (2006), los proyectos IS están creciendo en tamaño, y cada vez resulta más complicado cumplir con los requisitos, y la entrega ha de ser realizada más rápidamente. Por lo tanto, disponer de indicadores de eficiencia y eficacia a diversos niveles de medición es una necesidad diaria en las organizaciones del sector TIC. En esta línea, las medidas de productividad representan un indicador de cuán eficiente es el proceso productivo ejecutado a un nivel específico de análisis en una organización. Por ello, la medición de la productividad es una necesidad organizacional.

En IS, la medición de la productividad está ligada a la estimación y planificación de proyectos. Esto se debe a que una vez conocida la productividad en anteriores proyectos es posible inferir estimaciones de futuros proyectos, teniendo en cuenta las entradas que van a ser necesarias y las salidas que van a ser elaboradas en un determinado entorno de trabajo. La investigación sobre medidas de productividad en IS está principalmente centrada en la productividad a nivel de proyecto y en niveles superiores de medición (organización, sector, industria...), y existe escasa la literatura

sobre medidas de productividad a niveles inferiores (a nivel equipos o a individual) (Petersen, 2011).

A nivel de puesto de trabajo en IS, las medidas de productividad se han centrado en medir la productividad de la puesta en servicio de un determinado proyecto, bien mediante la medición de la salida con líneas de código o con puntos de función desarrollados y considerando como única entrada las horas hombre, u otra unidad de tiempo (Hernández-López, Colomo-Palacios, & García-Crespo, 2013). Estas medidas no representan la variedad de puestos de trabajo existentes en IS, ya que sólo miden la tarea de codificación, dejando a un lado las tareas de análisis y diseño, así como las tareas de gestión y mantenimiento de los proyectos realizados, pese a su influencia en la productividad, en la calidad y en los costes (Jones, 1981).

Los puestos de trabajo en IS encajan dentro de la definición de trabajadores del conocimiento. La medición de la productividad de estos trabajadores es un gran reto (Drucker, 1999). No sólo la propia medición, sino la definición de productividad desde un punto de vista de dichos puestos de trabajo que poco tienen que ver con los puestos de trabajo manufactureros (Nachum, 1999). Actualmente, como definición de productividad se utiliza la visión de la productividad como sinónimo de eficiencia y por lo tanto para su medida se emplean ratios entre salidas y entradas (Tangen, 2005). Sin embargo, es necesario adaptar la definición de productividad al nivel de puesto de trabajo para este tipo de trabajadores y por consiguiente elaborar nuevas medidas de productividad que se ajusten a dicha definición. Pese a las dificultades para medir la productividad de los trabajadores del conocimiento (Ramirez & Nembhard, 2004) es necesario crear nuevas medidas de productividad aplicables a los puestos de trabajo en los proyectos de desarrollo de software, principalmente en aquellos que no sólo se generen líneas de código o funcionalidad (Pfleeger, 2008).

Además, hay que tener en cuenta que antes de establecer nuevas medidas de productividad, es necesario identificar qué debe ser incluido en dichas medidas (Gummesson, 1992). En IS, varias características continúan sin estar incluidas en las

medidas de productividad a pesar de influir en los resultados. Por ejemplo, la reutilización no suele ser considerada a la hora de medir la productividad. Si consideramos la reutilización, y se mide la productividad por las líneas de código como salida, el resultado puede estar corrompido, incluso si se mide por la funcionalidad, ya que podría haber sido reutilizada o haberse sobretrabajado para hacer una funcionalidad reutilizable. Así pues, teniendo en cuenta estas indicaciones, y con el objetivo de crear una medida de productividad aplicable a los puestos de trabajo en los proyectos de desarrollo de software, es necesario obtener una definición de productividad que permita posteriormente distinguir los factores, entradas y salidas susceptibles de ser medidos. Desde un punto de vista más general, son necesarias medidas para cuantificar la productividad a lo largo de todas las fases del proceso productivo en IS. Solucionar estos problemas es de vital importancia dado que no todas las salidas de las actividades de IS son SLOC o PF o una medida *proxy* de estas medidas, y por otro lado, no todas las entradas son tiempos y esfuerzos. Así pues, son necesarias medidas que consideren otros elementos que no han sido tradicionalmente medidos (Petersen, 2011).

Como resumen, en la Tabla 1 se muestran los problemas identificados en esta investigación y las soluciones propuestas para cada uno de ellos. Estos problemas identificados constituirán la base a partir de la cual se definen los objetivos de esta tesis doctoral.

Problema	Solución propuesta
<p>P1: Identificar los recursos (entradas) y productos/servicios (salidas) susceptibles de ser considerados en la medición de la productividad.</p> <p>Las entradas que habitualmente se usan en las medidas de productividad se enfocan a la medición del tiempo y el esfuerzo; y las salidas se centran en la medición de la cantidad de código fuente o de la funcionalidad entregada. No obstante es razonable pensar que existen otros recursos empleados, y otros productos/servicios generados en los proyectos de desarrollo de software que podrían ser medidos, y por tanto, considerados en la medición de la productividad.</p>	<p>Se propone realizar un estudio para conocer la existencia de otras entradas, además del esfuerzo y del tiempo, y otras salidas, además del código fuente y la funcionalidad. Y analizar si existen diferencias entre cada puesto de trabajo en cuanto al uso de los recursos y la producción de productos/servicios.</p>
<p>P2: Definir medidas de productividad por puesto de trabajo.</p> <p>Cada puesto de trabajo tiene un conjunto de características que lo diferencia de los demás, por ende, las medidas de productividad a nivel de puesto de trabajo deberían tener en cuenta estas diferencias y ajustarse a la realidad de cada puesto de trabajo.</p>	<p>Se propone crear medidas de productividad específicas para cada puesto de trabajo incluido en la validación de forma que las medidas tengan en cuenta las características inherentes a cada puesto. Además, estas medidas tendrán en cuenta las entradas, salidas y factores específicos de cada puesto de trabajo.</p>
<p>P3: Elaborar nuevas medidas de productividad a nivel de puesto de trabajo.</p> <p>Las medidas de productividad a nivel de puesto de trabajo en IS parten de las medidas empleadas a niveles superiores de medición (proyecto, organización...) por lo que no han sido generadas mediante una aproximación específica para generar medidas de productividad en este nivel más específico y detallado.</p>	<p>Se propone validar las medidas creadas y compararlas con las medidas de productividad habitualmente utilizadas para comparar la eficacia de las medidas. Además, se propone un método de investigación que en sí mismo es una solución para la creación de nuevas medidas de productividad de otros puestos de trabajo.</p>

Tabla 1. Problemas identificados - Solución propuesta

1.3. Objetivos de investigación

El objetivo de investigación de esta tesis es **elaborar nuevas medidas de productividad para los puestos de trabajo en proyectos de desarrollo de software partiendo de las entradas y salidas previamente identificadas, y considerando los factores conocidos y aceptados en la literatura**. Para conseguir este objetivo es necesario identificar previamente las entradas que se utilizan y las salidas que se producen en el proceso productivo a medir, y los factores que afectan a los resultados de la medida. Con respecto a los factores, su identificación puede considerarse realizada a partir de varias investigaciones recientes (Paiva, Barbosa, Lima, & Albuquerque, 2010; Trendowicz & Münch, 2009; Wagner & Ruhe, 2008b) al menos de forma global aunque en cada organización y entorno pueden variar ya que dependen de las condiciones del puesto de trabajo, del trabajador, y de otras variables (Paiva, et al., 2010). Una vez definidos los factores que afectan a la productividad es necesario conocer las entradas y salidas que pueden ser utilizadas en las medidas de productividad. Y una vez conocidas las entradas y salidas potencialmente utilizables es posible elaborar nuevas medidas de productividad que se ajusten a la realidad de estos puestos de trabajo. Así pues, una vez conocidos los factores, entradas y salidas, la presente tesis doctoral tiene como objetivo final la construcción de nuevas medidas de productividad a nivel de puesto de trabajo en los proyectos de desarrollo de software. Los puestos de trabajo objeto de estudio son los siguientes: jefe de proyecto, consultor, analista, y programador; estos cuatro puestos están definidos en Métrica 3 (2000), y se excluye del objeto de estudio los puestos que se encuadran dentro del perfil directivo.

Para cumplir el objetivo de investigación de la presente tesis doctoral es necesario alcanzar los siguientes sub-objetivos:

- Obtener una definición de productividad a nivel de puesto de trabajo en los proyectos de desarrollo de software.

- Identificar las entradas que se utilizan a nivel de puesto de trabajo en los proyectos de desarrollo de software.
- Identificar las salidas que se producen a nivel de puesto de trabajo en los proyectos de desarrollo de software.
- Identificar las medidas de productividad que se utilizan a nivel de puesto de trabajo en los proyectos de desarrollo de software.
- Elaborar nuevas medidas de productividad a nivel de puesto de trabajo en los proyectos de desarrollo de software utilizando las entradas y salidas identificadas, y teniendo en cuenta los factores identificados.
- Evaluar si las nuevas medidas mejoran la medición de la productividad a nivel de puesto de trabajo con respecto a las identificadas previamente.

1.4. Hipótesis de investigación

De forma complementaria a los sub-objetivos de investigación de esta tesis, se plantean una serie de preguntas de investigación basadas en el estudio previo del estado de la cuestión y la experiencia del doctorando en diversos proyectos de desarrollo de software. Estas preguntas de investigación son las siguientes:

- **Pregunta 1.** ¿Existen otras entradas medibles en la medición de la productividad a nivel de puesto de trabajo que no sean las horas de trabajo ni el esfuerzo humano?
- **Pregunta 2.** ¿Son distintas las entradas por puesto de trabajo en los proyectos de desarrollo de software?
- **Pregunta 3.** ¿Existen otras salidas medibles en la medición de la productividad a nivel de puesto de trabajo que no sean las líneas de código ni la funcionalidad?
- **Pregunta 4.** ¿Son distintas las salidas por puesto de trabajo en los proyectos de desarrollo de software?

- **Pregunta 5.** ¿Son eficaces las actuales medidas de productividad a nivel de puesto de trabajo en los proyectos de desarrollo de software para medir la productividad de todos los puestos?
- **Pregunta 6.** ¿Es posible utilizar medidas que combinen varios elementos (entradas y/o salidas) para definir nuevas medidas de productividad a nivel de puesto de trabajo en los proyectos de desarrollo de software?

A partir de estas preguntas de investigación es posible definir las siguientes hipótesis de investigación:

- **Hipótesis 1.** En los puestos de trabajo involucrados en la ejecución de proyectos de desarrollo de software se emplean otras entradas, además del tiempo y el esfuerzo.
- **Hipótesis 2.** Las entradas utilizadas son distintas para cada puesto de trabajo involucrado en la ejecución de proyectos de desarrollo de software.
- **Hipótesis 3.** En los puestos de trabajo involucrados en la ejecución de proyectos de desarrollo de software se producen otras salidas, además de líneas de código y funcionalidad.
- **Hipótesis 4.** Las salidas producidas son distintas para cada puesto de trabajo involucrado en la ejecución de proyectos de desarrollo de software.
- **Hipótesis 5.** Las medidas más utilizadas de productividad a nivel de puesto de trabajo en los proyectos de desarrollo de software tienen una eficacia limitada para medir la productividad real de los trabajadores.
- **Hipótesis 6.** Es posible medir de forma más eficaz la productividad de los puestos de trabajo en los proyectos de desarrollo de software con nuevas medidas que combinen varios elementos: entradas, salidas y factores.

Las cuatro primeras hipótesis serán contrastadas en el enfoque del problema (ver el capítulo 3 Enfoque del problema) mediante una fase de investigación cualitativa y una posterior cuantitativa. Una vez finalizado el enfoque del problema, será posible contrastar las dos últimas hipótesis tras realizar la validación de la solución aportada.

El proceso diseñado para aproximarse a la solución, junto con el momento en el que se contrastan las hipótesis, se detalla de forma más precisa en el siguiente apartado.

Hipótesis	Problemas de investigación relacionados
Hipótesis 1. En los puestos de trabajo involucrados en la ejecución de proyectos de desarrollo de software se emplean otras entradas, además del tiempo y el esfuerzo.	P1: Recursos (entradas) y productos/servicios (salidas) susceptibles de ser considerados en la medición de la productividad.
Hipótesis 2. Las entradas utilizadas son distintas para cada puesto de trabajo involucrado en la ejecución de proyectos de desarrollo de software.	P2: Medidas de productividad por puesto de trabajo.
Hipótesis 3. En los puestos de trabajo involucrados en la ejecución de proyectos de desarrollo de software se producen otras salidas, además de líneas de código y funcionalidad.	P1: Recursos (entradas) y productos/servicios (salidas) susceptibles de ser considerados en la medición de la productividad.
Hipótesis 4. Las salidas producidas son distintas para cada puesto de trabajo involucrado en la ejecución de proyectos de desarrollo de software.	P2: Medidas de productividad por puesto de trabajo.
Hipótesis 5. Las medidas más utilizadas de productividad a nivel de puesto de trabajo en los proyectos de desarrollo de software tienen una eficacia limitada para medir la productividad real de los trabajadores.	P1: Recursos (entradas) y productos/servicios (salidas) susceptibles de ser considerados en la medición de la productividad. P2: Medidas de productividad por puesto de trabajo. P3: Nuevas medidas de productividad a nivel de puesto de trabajo.
Hipótesis 6. Es posible medir de forma más eficaz la productividad de los puestos de trabajo en los proyectos de desarrollo de software con nuevas medidas que combinen varios elementos: entradas, salidas y factores.	P1: Recursos (entradas) y productos/servicios (salidas) susceptibles de ser considerados en la medición de la productividad. P2: Medidas de productividad por puesto de trabajo. P3: Nuevas medidas de productividad a nivel de puesto de trabajo.

Tabla 2. Hipótesis y problemas de investigación relacionados

1.5. Aproximación a la solución

La solución que plantea la presente tesis doctoral se basa en la aplicación de un método de medición de productividad existente para medir la productividad de varios puestos de trabajo dentro de los proyectos de desarrollo de software. Además, se comparan los resultados obtenidos mediante la aplicación de este método con los métodos tradicionalmente usados. Para llegar a esto, se ha llevado un proceso investigador que en si mismo constituye un marco de trabajo para posteriores estudios de medición de productividad en otros puestos dentro de IS o en otras áreas de conocimiento (ver el apartado 1.6 Método de investigación). De este modo se podrá disponer de un marco de trabajo para analizar los elementos que componen la productividad de trabajadores del conocimiento, y establecer medidas de productividad más eficaces. En concreto, la solución propuesta se basa en la utilización de DEA (*Data Envelopment Analysis*) para medir la productividad de varios puestos de trabajo dentro de IS.

El método DEA fue creado de forma preliminar en los años 50 por Farrell (1957) y definido finalmente por Charnes, Cooper y Rhodes a finales de los 70 (1978) añadiendo rendimientos constantes a escala, y posteriormente suponiendo rendimientos variables a escala por Banker, Charnes y Cooper (1984) lo que permite calcular eficiencias de escala. DEA es un método de programación lineal no paramétrico que no fija una forma explícita sobre los datos y crea una función de producción mediante la inclusión en el análisis de varias categorías de entradas (recursos) y salidas (productos o servicios). Para ello utiliza unidades de análisis denominadas *Decision Making Unit* (DMU). Farrell propuso que la eficiencia de una DMU consiste en dos componentes: “eficiencia técnica”, que refleja la habilidad para obtener el máximo *output* para un conjunto dado de *inputs*, y la “eficiencia redistributiva”, que refleja la habilidad para usar los *inputs* en las proporciones óptimas, dados sus respectivos precios; ambas medidas se combinan para obtener la “eficiencia económica”. Este método compara cada medición con el resto de mediciones, y por lo tanto no se obtiene un resultado

universal. En IS, el método DEA ha sido utilizado con múltiples propósitos. Por ejemplo, ha sido utilizado para analizar los factores que influyen en el tamaño de los componentes en Orientación a Objetos y de la documentación de código fuente (Pendharkar, 2006), para analizar los rendimientos a escala en el desarrollo de software (Banker & Kemerer, 1989), y para evaluar la relación entre los PF post implementación y el esfuerzo de desarrollo correspondiente (Asmild, Paradi, & Kulkarni, 2006). A nivel de trabajadores, ha sido utilizado para examinar la productividad, eficiencia y eficacia de un equipo de diseño en ingeniería (Paradi, Smith, & Schaffnit-Chatterjee, 2002). El método DEA cubre cinco de las 13 dimensiones destacadas por Ramírez y Nembhard en su estudio sobre la medición de la productividad para los trabajadores del conocimiento (2004). Cabe mencionar que los métodos encontrados que más dimensiones cubren lo hacen con cinco, y sólo dos modelos teóricos cubren seis dimensiones. En concreto el método DEA cubre las dimensiones de *Cantidad*, *Coste y/o Rentabilidad*, *Eficiencia*, *Efectividad*, y *Éxito del Proyecto*. En comparación, medir la productividad mediante PF solo cubre 2 dimensiones, *Cantidad* y *Calidad*, según Ramírez y Nembhard (2004), por lo que medir la productividad mediante DEA es, al menos basado en estas dimensiones, más eficaz.

Teniendo en cuenta las particularidades de la metodología seleccionada, **en esta tesis se utiliza DEA para construir nuevas medidas de productividad a nivel de puesto de trabajo en proyectos de desarrollo software**. Para alcanzar este propósito, se ha construido y ejecutado el método de investigación que se detalla en el siguiente apartado.

1.6. Método de investigación

El proceso de investigación para elaborar nuevas medidas de productividad a nivel de puesto de trabajo vinculados con la ejecución de proyectos de desarrollo software requiere la realización de unos pasos previos (ver Ilustración 2). El proceso se ha

divido en tres etapas: (1) revisión del estado de la cuestión, (2) enfoque del problema, y (3) desarrollo de nuevas medidas.

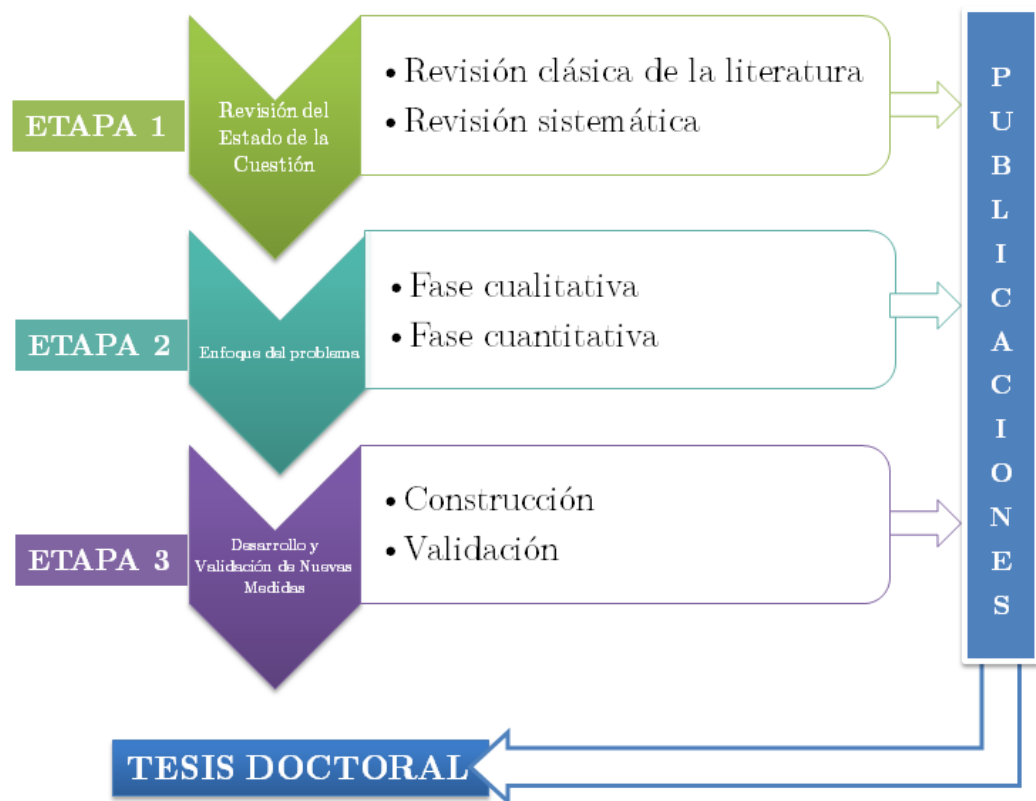


Ilustración 2. Aproximación a la solución

En la primera etapa - Revisión del Estado de la Cuestión - (ver Ilustración 3), se analizará la literatura para conocer el estado de la cuestión referente a medidas de productividad a nivel de puesto de trabajo dentro de IS y en especial dentro de los proyectos de desarrollo de software. Para ello, se realizará una revisión clásica de la literatura, es decir, un análisis de referencias de varios estudios para cada uno de los temas a analizar y se profundizará en las referencias de estos estudios; esta operativa se repetirá hasta que no aparezcan estudios relevantes para los objetivos de investigación. De forma adicional, se realizará una revisión sistemática de la literatura (Kitchenham, 2004) que permitirá dar respuesta precisa mediante el empleo de un protocolo de revisión. Este tipo de revisión permite que el trabajo realizado pueda ser duplicado, dado que todos los pasos seguidos son transparentes y documentados. Estas

características no están presentes en la revisión clásica en la que no existe ningún proceso ni documentación referente a la metodología empleada (Kitchenham, 2007).

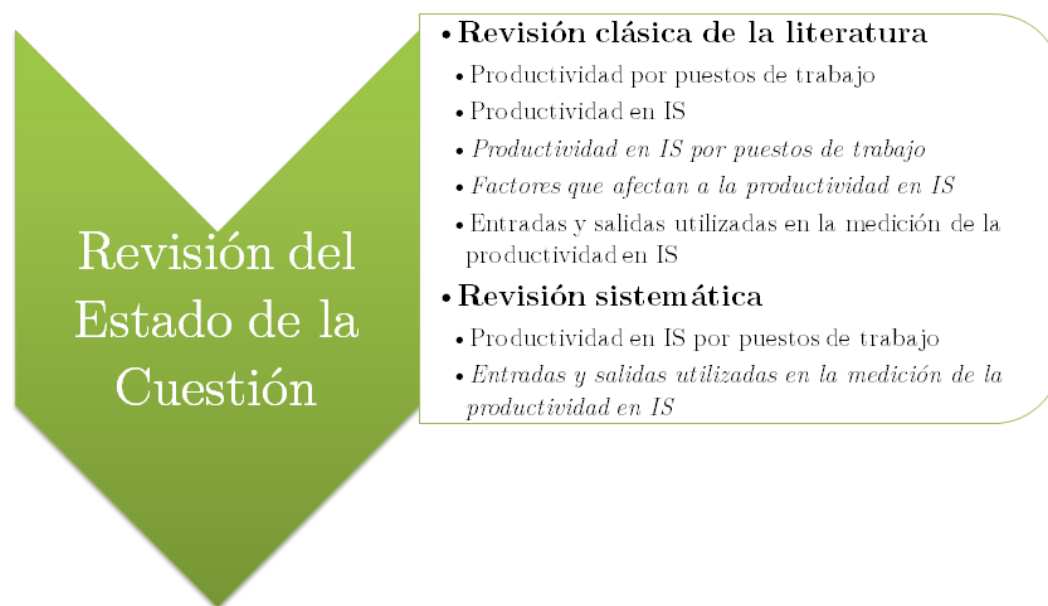


Ilustración 3. Aproximación a la solución - Etapa 1 (Revisión del Estado de la Cuestión)

Una vez el estado de la cuestión ha sido analizado, se dará paso a la segunda etapa - Enfoque del Problema - (ver Ilustración 4) en la que se identificarán las entradas y salidas objeto de medición para las nuevas medidas de productividad. Además, se obtendrá una definición de productividad en varios niveles de medición (organización, proyecto, y puesto de trabajo). A su vez, se identificarán los factores que afectan a la motivación y satisfacción laboral de los trabajadores en IS. Para identificar las entradas y salidas, se realizará una fase cualitativa de carácter exploratorio (Patton, 1980; Strauss & Corbin, 1990) de las entradas que utilizan los trabajadores en IS y las salidas que producen, de modo que será posible contrastar las cuatro primeras hipótesis planteadas (H1 a H4) con información cualitativa. Pese a que los factores que afectan a la productividad son de gran importancia, su identificación no será realizada en la presente tesis doctoral ya que existen varias investigaciones recientes que tratan esa cuestión – incluso en forma de revisiones sistemáticas de la literatura (Paiva, et al., 2010; Trendowicz & Münch, 2009; Wagner & Ruhe, 2008b). No obstante, en la fase cualitativa, se analizará, desde el punto de vista de los

trabajadores, la influencia de los factores que ellos consideren influyentes en su productividad. Esta labor, será posible gracias al empleo de una metodología de investigación cualitativa basada en entrevistas a trabajadores en el área IS. Una vez realizada esta fase cualitativa será posible utilizar métodos cuantitativos, en una segunda fase, basados en el empleo de cuestionarios y análisis estadístico para contrastar las cuatro primeras hipótesis planteadas (H1 a H4).

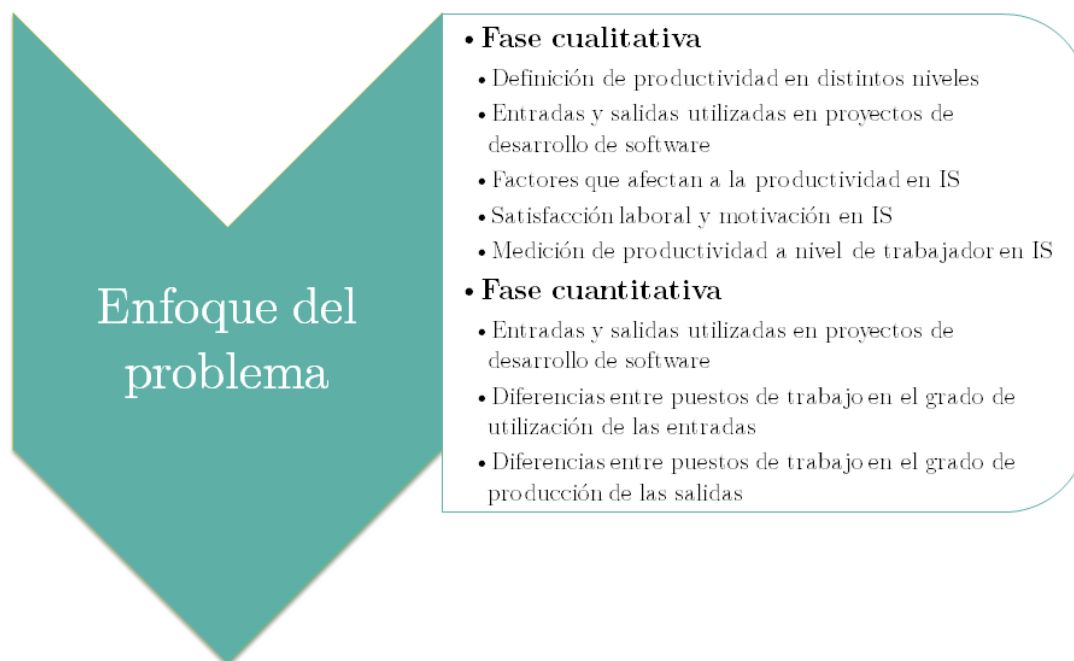


Ilustración 4. Aproximación a la solución - Etapa 2 (Enfoque del problema)

En la tercera etapa - Desarrollo y Validación de Nuevas Medidas - (ver Ilustración 5), se procederá a desarrollar y validar nuevas medidas una vez hayan sido identificadas las entradas y salidas, y conociendo si existen diferencias entre puestos en relación al uso de las entradas y/o de producción de las salidas. Además, estas medidas tendrán en cuenta las entradas y salidas identificadas en las anteriores tareas y los factores identificados en la literatura. Finalmente, estas medidas serán evaluadas para contrastar su eficacia en la medición de la productividad por puestos de trabajo dentro de los proyectos de desarrollo de software. Una vez finalizada la construcción y prueba de las nuevas medidas, será posible contrastar las dos últimas hipótesis planteadas (H5 y H6).

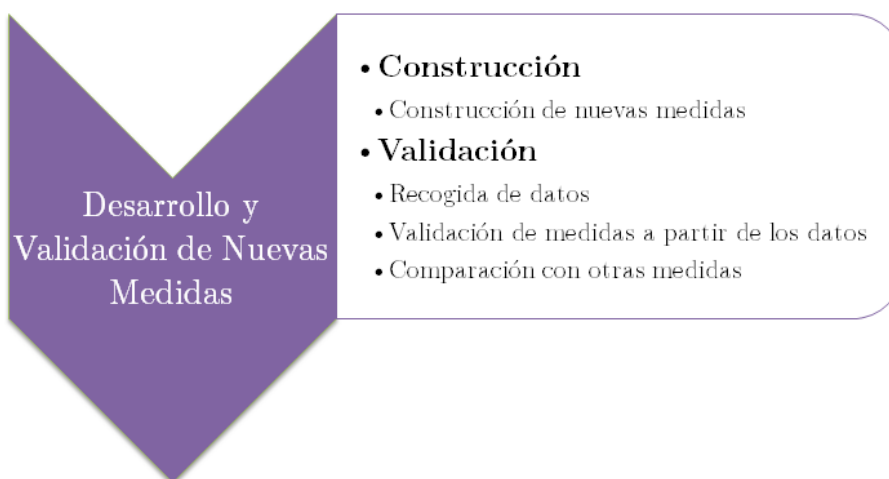


Ilustración 5. Aproximación a la solución - Etapa 3 (Desarrollo y Validación de Nuevas Medidas)

De cada una de estas etapas se publicarán los resultados obtenidos en revistas y congresos de diverso prestigio. Inicialmente se plantea realizar dos publicaciones por cada una de las etapas, aunque podrían ser más si el material obtenido permitiera más publicaciones. De este modo, la construcción del presente documento, que plasma la tesis doctoral ejecutada, será, en parte, una recopilación de las publicaciones realizadas a partir de los resultados de cada una de las etapas. Para ver las publicaciones que han sido realizadas ver el Anexo F - Publicaciones relacionadas con la tesis.

1.7. Aportaciones de la investigación

Las aportaciones de la presente investigación se dividen en dos partes. Por un lado las aportaciones a la propia investigación, y por otro lado a las organizaciones. Con respecto a la investigación, la metodología de investigación empleada puede ser reutilizada en otras áreas de conocimiento en las cuales sea necesario establecer nuevas medidas de productividad. Además, puede ser reutilizada para construir nuevas medidas de productividad para otros puestos de trabajo dentro de IS que no han sido tratados en la presente tesis doctoral. Así pues, la presente tesis doctoral en si misma representa un marco metodológico

para analizar, definir y construir nuevas medidas de productividad para puestos de trabajo. Con respecto a las organizaciones, las aportaciones son las siguientes:

- **Aumento de la eficacia de la medición de la productividad de los trabajadores.** Las medidas de productividad propuestas permiten utilizar múltiples entradas y múltiples salidas. La eficacia de la medición es mayor en cuanto a que refleja con mayor realismo la casuística de los puestos de trabajo bajo estudio. En estos puestos de trabajo se utilizan más recursos además del esfuerzo y se producen más salidas que el código fuente y la funcionalidad, y éstas no son *proxies* de las otras entradas y/o salidas. Así pues, las medidas de productividad que solo utilizan una entrada y/o una salida no son completas y su eficiencia es menor que la de la solución aportada.
- **Ajuste de la medida de productividad a cada entorno** mediante la variación de las entradas y salidas incluidas en la medida. De este modo, cada organización puede ajustar la medida considerando las entradas y salidas en base a sus necesidades y las características de su entorno. A su vez se eliminan los problemas de ajuste de algunos modelos de medición a determinados entornos.
- **Detección de ineficiencias de utilización de entradas y/o de producción de salidas.** Mediante las medidas de productividad propuestas es posible identificar las ineficiencias existentes, tanto en las entradas como en las salidas, lo que permite incidir en ellas, corregirlas, y, por consiguiente, aumentar la productividad.
- **Comparación de resultados con respecto al mejor resultado** a través de los resultados obtenidos en la utilización de la medida de productividad. La medida aportada compara los resultados de productividad con el resto de resultados, con lo cual se obtiene una clasificación, es decir, la productividad se compara siempre con el mejor,

no con el promedio. De este modo, y teniendo en cuenta que es posible detectar las ineficiencias, la solución aporta una herramienta de *benchmarking* interno a la organización.

1.8. Validez de la solución

La validez de la solución aportada se divide en dos secuencias, por un lado las hipótesis (H1 a H4) se comprueban mediante contraste estadístico, y las dos últimas hipótesis (H5 y H6) mediante la utilización de la metodología empírica de casos de estudio. Así pues:

1. Secuencia I: En los puestos de trabajo vinculados con los proyectos de desarrollo de software se utilizan otros recursos además del tiempo y el esfuerzo de los trabajadores, y se generan otras salidas además del código fuente y la funcionalidad. Además, la utilización de las entradas y la producción de las salidas varía en función del puesto de trabajo.
2. Secuencia II: La eficacia de las medidas de productividad habitualmente utilizadas para medir la productividad a nivel de puestos de trabajo en los puestos vinculados con los proyectos de desarrollo de software puede ser mejorada.

Secuencia I: En los puestos de trabajo vinculados con los proyectos de desarrollo de software se utilizan otros recursos además del tiempo y el esfuerzo de los trabajadores, y se generan otras salidas además del código fuente y la funcionalidad. Además, la utilización de las entradas y la producción de las salidas varían en función del puesto de trabajo.

El objetivo de la Secuencia I es contrastar si en los puestos de trabajo de jefe de proyecto software, consultor, analista, y programador: (1) se utilizan otras entradas además del tiempo y el esfuerzo, (2) se producen otras salidas además de código fuente y funcionalidad, y (3) existen diferencias significativas en el grado de utilización de las

entradas, y en la producción de salidas en función del puesto de trabajo. En esta secuencia se contrastan las hipótesis H1, H2, H3 y H4 de la presente tesis doctoral.

El propósito de esta secuencia es:

1. Contrastar la existencia de otras entradas a parte del tiempo y el esfuerzo en los puestos de trabajo bajo estudio.
2. Contrastar la existencia de otras salidas a parte del código fuente y la funcionalidad en los puestos de trabajo bajo estudio.
3. Contrastar si en los puestos de trabajo bajo estudio existen diferencias en el grado de utilización de las entradas.
4. Contrastar si en los puestos de trabajo bajo estudio existen diferencias en el grado de producción de las salidas.

Para ello, se realizará una fase de investigación cualitativa previa para conocer las entradas y salidas utilizadas desde el punto de vista de los trabajadores. En esta fase se utilizará una metodología cualitativa basada en entrevistas para la toma de información, y en análisis de contenido para analizar los datos y obtener los resultados. Tras esta fase será posible contrastar las hipótesis (H1 a H4) con información cualitativa. Además, los resultados permitirán disponer de información suficiente para continuar con la siguiente fase. Posteriormente, se realizará una fase cuantitativa para contrastar las cuatro hipótesis planteadas con información estadística. En esta fase se utilizará un cuestionario como método de recogida de información y un método estadístico no paramétrico para contrastar las hipótesis planteadas.

Secuencia II: La eficacia de las medidas de productividad habitualmente utilizadas para medir la productividad a nivel de puesto de trabajo en los puestos vinculados con los proyectos de desarrollo de software puede ser mejorada.

El objetivo de la Secuencia II es construir y validar nuevas medidas de productividad para los puestos de trabajo de ingeniero de software.

El propósito de esta secuencia es:

1. Analizar la eficacia de las medidas de productividad habitualmente utilizadas para medir la productividad de los puestos de trabajo programador, analista, consultor y jefe de proyecto.
2. Construir nuevas medidas de productividad teniendo en cuenta los resultados previos.
3. Validar la eficacia de las medidas creadas.
4. Comparar los resultados obtenidos con las medidas habitualmente utilizadas.

Para conseguir el objetivo de esta secuencia se llevará a cabo una revisión sistemática de la literatura junto con una revisión clásica para conocer las medidas habitualmente utilizadas en la medición de la productividad de los trabajadores en IS. Posteriormente, se construirán nuevas medidas de productividad teniendo en cuenta los resultados obtenidos en la Secuencia I y el análisis de la literatura. Una vez construidas las medidas, estas serán validarlas y compararlas con las medidas previamente identificadas en la literatura.

1.9. Estructura de la tesis doctoral

La presente tesis doctoral está estructurada de la siguiente manera:

- **Introducción:** Es este capítulo, se presenta el marco de la investigación. Se define el problema en función de la situación actual dentro de este contexto, y se recoge la motivación de la investigación. Además, se describen los objetivos e hipótesis de investigación, y se expone de forma breve una aproximación a la solución. A continuación, se detallan las aportaciones a la

investigación y por último se describe cómo se realiza la validez de la solución propuesta en esta investigación.

- **Revisión crítica del estado de la cuestión:** En este capítulo se presenta el estado de la cuestión realizando una revisión crítica de las investigaciones existentes. Además, se presenta una revisión sistemática de la literatura sobre las medidas de productividad a nivel de puesto de trabajo en IS, junto con las entradas y salidas que en estas medidas se utilizan.
- **Enfoque del problema:** En este capítulo se presenta el estudio realizado (mediante una fase cualitativa y otra cuantitativa) que tiene por objetivo contrastar las cuatro primeras hipótesis de investigación planteadas.
- **Solución propuesta:** En este capítulo se especifica la solución propuesta mediante la cual se mide la productividad de los puestos de trabajo relacionados directamente con los proyectos de desarrollo de software.
- **Validación:** En este capítulo se describe la planificación, ejecución y el análisis de los datos obtenidos en la validación experimental realizada. Además, se contrastan las dos últimas hipótesis de investigación planteadas.
- **Conclusiones y futuras líneas de investigación:** En este capítulo se muestran las conclusiones de la presente tesis doctoral, junto con posibles líneas futuras de investigación.
- **Referencias:** En este capítulo se encuentra la lista de referencias a los principales trabajos consultados y utilizados para la realización de esta tesis doctoral.
- **Anexos:** En este último capítulo se encuentran varios elementos que, debido a su gran tamaño, su aporte e interés, y su carácter complementario, han sido agrupados dentro de este capítulo para no dificultar la lectura de la tesis doctoral.

2 Revisión del estado de la cuestión

2.1. Introducción

Los orígenes del término “productividad” se remontan al siglo XVIII (Quesnay, 1766). Sin embargo, hasta mediados del siglo pasado, su definición era confusa. Tradicionalmente, ha sido definida como el ratio de unidades producidas por unidad de entrada; es decir, como sinónimo de la eficiencia en el uso de la entrada para producir unas salidas (Jefferys, et al., 1954). Esta definición encaja perfectamente en el paradigma de la fabricación dado que está basado en cantidades de unidades de medida estandarizadas y claramente identificadas, pero no encaja en nuevos entornos tales como IS, en los que el proceso productivo no tiene las mismas características. En estos nuevos entornos se producen y utilizan activos tangibles e intangibles, las tareas no siempre son observables, ciertos elementos son difíciles de medir, el cliente está presente en el proceso productivo, etc. Es por ello que en las industrias de servicios, la productividad debe ser vista como un componente del rendimiento, no como un sinónimo de éste (Sink, Tuttle, & DeVries, 1984). Además, como Grönroos y Ojasalo (2004) concluyen, la noción de la productividad en las industrias de servicios tiene el efecto combinado de la efectividad del proveedor de servicios que gestiona la rentabilidad de sus recursos y procesos de producción (eficiencia interna), y la calidad percibida de sus servicios (eficiencia externa), lo que hacen de la productividad un concepto diferente en comparación con el concepto tradicional de productividad. Por

otro lado, la medición de la productividad tiene por objeto ser comparada con anteriores medidas de productividad, y no como una unidad de resultado, es decir, las medidas de productividad deben ser utilizadas para compararlas a lo largo del tiempo, mientras que el rendimiento representa una medida temporal. Siguiendo esta puntualización, el valor de las medidas de productividad se encuentra en la capacidad de gestionar y controlar los recursos para alcanzar un uso más eficiente de los mismos (Fitzgerald & Moon, 1996). Además, como Nachum (1999) sostuvo, el objetivo principal de la medición de la productividad es la mejora de la productividad; y, las mejoras de la productividad deben estar reflejadas en mejoras en el retorno de la inversión (ROI, *Return On Investments*). De este modo, una medida de productividad apropiada aporta una herramienta de pronóstico adecuada para lograr esa mejora de la productividad (Nachum, 1999). Es por todo esto por lo que la mejora de la productividad en IS no puede ser realizada sin una medición eficaz de la productividad. Tal y como Anselmo y Ledgard (2003) indicaron siguiendo la afirmación de Lord Kelvin¹, la mejora de la productividad en el desarrollo de software no puede ser realizada sin una medición de la productividad.

En IS, el interés científico por la medición de la productividad puede decirse que comenzó a finales de los años 70. En sus orígenes, los objetos de estudio fueron principalmente la medición de la productividad en las actividades de programación (Chrysler, 1978) y los proyectos de software (Albrecht, 1979). Además, en esa época se empezaron a considerar los factores que afectan a la productividad (Thadhani, 1984; Vessey & Weber, 1983). Con el paso del tiempo, y dado que la medición de la productividad tiene como uno de sus objetivos comparar productividades, fueron apareciendo investigaciones que realizaban comparaciones a diferentes niveles de análisis: sector, organización, proyecto... (Cusumano & Kemerer, 1990; Maxwell, Wassenhove, & Dutta, 1996). De igual forma, el estudio de los factores continuó en las

¹ “When you can measure what you are speaking about, [...] you know something about it; but when you cannot measure it, [...] your knowledge is of a meager and unsatisfactory kind [...]”

siguientes décadas (Scacchi, 1994). Por otro lado, se fueron incluyendo elementos a la medición de la productividad tales como la reutilización (Banker & Kauffman, 1991; Gaffney, 1989) o la orientación a objetos (Purao & Vaishnavi, 2003). No obstante, la utilización de estos elementos no parece haber sido aceptada ampliamente ya que las medidas de productividad continúan utilizando los estudios de los años 70 y 80 como marco de trabajo (Bellini, Pereira, & Becker, 2008; Kitchenham, 2010; Pfleeger, 2008). La finalidad que persiguen estas medidas es medir la eficiencia con la que se termina un proyecto software, bien mediante la medición de las líneas de código desarrolladas (p. ej. contando las SLOC) o de la funcionalidad entregada (p. ej. midiendo el producto en PF). Este tipo de medición permite emplear los resultados obtenidos para estimar futuros proyectos software utilizando alguno de los métodos disponibles ya que se utilizan las mismas medidas, SLOC o PF (Boehm, Abts, & Chulani, 2000). Así pues, y dado que uno de los retos clásicos de la industria del software es la entrega en tiempo de los proyectos (Boehm & Ross, 1989), la utilización de medidas de productividad que tengan por objeto medir la eficiencia de entrega no ha de resultar extraño (Dale & van der Zee, 1992). Este es el caso de las medidas desarrolladas en los años 70 y 80 continúan empleándose. Sin embargo, estas medidas de productividad no miden todas las actividades de IS, lo que se traduce en una carencia de medición de la productividad a nivel de puesto de trabajo. De este modo, es necesario elaborar medidas de productividad para dicho nivel. Para ello, y siguiendo las recomendaciones de Gummesson (1992), antes de cualquier intento de medir la productividad, es necesario identificar qué debe ser considerado en dicha medida. Y para identificar qué debe ser considerado hay que distinguir los factores, entradas y salidas susceptibles de ser medidos.

En primer lugar, los factores que afectan a la productividad en IS son de diversa tipología y pueden organizarse en dos grandes categorías: por un lado los factores técnicos (producto, proceso, entorno de desarrollo) y por otro lado los factores *soft* (cultura organizacional, cultura de equipo, capacidades y experiencias, entorno, proyecto) (Wagner & Ruhe, 2008b). Otras tipologías clasifican los factores de forma

parecida (Santillo & Moretto, 2011; Scacchi, 1994). Aunque muchos de estos factores son ampliamente conocidos desde hace años - por ejemplo, la influencia de la experiencia (Boehm, 1981b; Walston & Felix, 1977) y el lenguaje de programación (entorno de desarrollo) - no está claro que su importancia realmente sea la que hace años se le adjudicó ya que las prácticas y procesos de trabajo así como las herramientas empleadas han evolucionado considerablemente (Paiva, et al., 2010). Además, es muy difícil determinar el grado de influencia y la valencia de los factores, ya que cada sujeto de estudio es casi imposible de replicar. Por ejemplo, Sommerville (2010) señala la importancia del entorno de trabajo mencionando que mediante la combinación de espacios individuales para el trabajo técnico para evitar distracción y espacios compartidos para comunicarse es posible aumentar la productividad, pero ¿en qué momento la reducción del espacio de trabajo reduce la productividad? (Jones, 1995). Lo que sí parece estar claro entre los investigadores es la existencia de estos dos grandes grupos de factores: “factores de personas” (*people factors*) y “factores tecnológicos” (*technical factors*), como se desprende de varias revisiones de la literatura sobre factores (ver p. ej.:Wagner & Ruhe, 2008b).

En segundo lugar, las entradas del proceso productivo que han sido utilizadas en la medición de la productividad se centran en unidades de tiempo, principalmente horas y días, y en unidades de esfuerzo, por ejemplo horas-hombre (Gómez, Oktaba, Piattini, & García, 2008; Ramirez & Nembhard, 2004). La utilización de estas entradas se puede deber a dos intereses. Por un lado, la entrega de un proyecto en el tiempo planificado, así como la puesta en el mercado del mismo es un factor importante para el éxito de un proyecto; muestra de este interés es el empleo de prácticas organizacionales tales como el *Global Software Development* (Herbsleb & Moitra, 2001). Por otro lado, la mayor partida de costes en las organizaciones productoras de software es el coste o esfuerzo en capital humano (Jørgensen & Shepperd, 2007); de modo que el interés en emplear esta entrada puede ser de índole económica. Pero, ¿realmente sólo se necesita tiempo y esfuerzo para producir software?, y si sólo se mide el tiempo ¿qué tiempo debería medirse: el empleado por el

trabajador para llevar a cabo sus tareas o el pagado por el empleador? De forma paralela y con el avance del tiempo, el coste de los equipos hardware se ha ido reduciendo y el coste de estos recursos es casi irrelevante en proyectos de gran tamaño en comparación con el coste en capital humano. Así pues, cualquier recurso de interés para la medición de la productividad estará relacionado directa o indirectamente con el capital humano. No obstante, el tiempo es sin lugar a duda uno de los principales recursos que está relacionado con el capital humano, ya que a mayor tiempo mayor empleo de mano de obra (Davenport & Prusak, 2000), sin embargo no parece ser la única entrada cuando sigue en el aire la visión de IS como una tarea artesanal (Boehm & Ross, 1989).

En tercer lugar, las salidas del proceso productivo han seguido la misma filosofía que las entradas: la entrega en tiempo. De este modo, se han medido principalmente las líneas de código o la funcionalidad, midiendo de esta forma el tamaño del producto desarrollado (Albrecht & Gaffney, 1983). Estas salidas tienen gran aceptación en las organizaciones pese a su limitada eficacia para medir las salidas producidas. Por un lado, la medición basada en PF es una forma de medir la funcionalidad que ha de ser desarrollada y por lo tanto de medir qué se entrega al cliente. Por otro lado, la basada en SLOC se limita a las instrucciones que codifican la funcionalidad desarrollada (Petersen, 2011). No obstante, estos dos tipos de salida no son los únicos que se producen en IS y se limitan exclusivamente a medir el tamaño del producto desarrollado. Una salida importante que queda fuera de estas salidas es la calidad, la cual afecta a la salida producida (eficacia) y al proceso productivo en sí mismo (eficiencia) (Al-Darrab, 2000). Además, se producen otras salidas, algunas relacionadas con la organización del trabajo (p. ej. alcance de hitos y finalización de tareas) o de tipo producto (p. ej. requisitos y diseños) que no siempre son medidas. De forma complementaria, hay características de las salidas que pueden afectar a la productividad y que no suelen ser medidas, tanto por su creación inicial como por su posterior uso, entre ellas destacan la reutilización (Anselmo & Ledgard, 2003) y la documentación (Boehm, 1987).

De forma adicional, el nivel de análisis debe ser tenido en cuenta cuando se mide la productividad en IS. Como niveles de medición se pueden señalar los siguientes: sector, organización, departamento, proyecto, unidad, equipo, y puesto de trabajo. Hay que destacar que no todas las medidas son eficaces a todos los niveles ya que las características y objetivos son distintos en cada uno de ellos. Así pues, debe haber una clara especificación del objetivo de la medida y las audiencias interesadas en los resultados de dichas medidas; esta información debería estar definida antes que cualquier otra información (Sink, et al., 1984). Es por ello por lo que a nivel de puesto de trabajo, la granularidad de los factores, entradas y salidas envueltas no es el mismo que a nivel de sector u organización para una medida eficaz que para ser utilizada como fuente de información en la mejora de ésta.

Además de estas consideraciones, hay que tener en cuenta que los puestos de trabajo vinculados con los proyectos de desarrollo de software encajan en el tipo de trabajador llamado trabajador del conocimiento (Drucker, 1959). Estos puestos de trabajo están caracterizados por una fuerte carga de requisitos intelectuales en lugar de requisitos manuales y físicos. En muchos casos, algunos de los elementos que producen son de naturaleza intangible. Así pues, medir la productividad de estos trabajadores representa un reto para la investigación (Drucker, 1999); en esta dirección algunos estudios han arrojado algo de luz pero no son concluyentes (Erne, 2011; Ramirez & Nembhard, 2004). Los trabajadores del conocimiento están presentes en casi todos los sectores productivos, su número va en aumento, y aunque la proporción con otros tipos de trabajadores es baja, su influencia en la economía es mayor (Davenport, 2005). En los proyectos de desarrollo software, teniendo en cuenta la tipología de puestos de trabajo empleados, se puede decir que todos los puestos encajan en la descripción de trabajadores del conocimiento. En el caso de los proyectos de desarrollo software, se desempeñan tareas tanto tangibles, por ejemplo la programación, como intangibles, p. ej. el diseño y el análisis (Lowry, 2010; Rombach, 1990). Así pues, a pesar de la dificultad para definir la productividad es necesario investigar nuevas medidas de productividad para estos trabajadores siguiendo la

máxima *"Si no se puede medir, no se puede gestionar"*², y por ende tampoco puede ser mejorado.

En este punto, es necesario preguntarse si las prácticas actuales de medición de la productividad en IS siguen procesos sistemáticos de definición y diseño como el planteado por Sink et al. (1984); y, si los factores, las entradas y las salidas están debidamente identificados en el nivel de medición requerido. Y por otro lado, si los factores, salidas y entradas ampliamente aceptados como la reutilización, la calidad, la experiencia, el conocimiento y las características del capital humano empleado están considerados en las medidas de productividad empleadas. Para obtener una visión de estos problemas, se presenta a continuación el estado de la cuestión referente a los siguientes puntos:

- Elementos necesarios para la medición de la productividad:
 - Definición de productividad.
 - Factores.
 - Entradas y salidas.
 - Motivación y satisfacción laboral.
- Cuestiones.
- Puestos de trabajo, con especial atención en los puestos relacionados con los proyectos de desarrollo de software, y los puestos definidos en Métrica 3 (2000).
- Medición de la productividad en IS, con especial atención en la medición a nivel de puesto de trabajo en los puestos relacionados con los proyectos de desarrollo de software.

Tras analizar estos puntos, se presenta la revisión sistemática de la literatura realizada sobre medidas de productividad a nivel de puesto de trabajo en ingeniería

² Traducción de "If we can't measure it, we can't manage it"

del software. Finalmente se resume en el estado de la cuestión para dar paso al enfoque de problema.

2.2. Elementos necesarios para la medición de la productividad

Antes de elaborar una medida de productividad es necesario identificar qué va a ser medido y qué influye en lo que va a ser medido. Para ello en esta revisión del estado de la cuestión van a ser identificados desde el punto de vista de la literatura científica actual, y teniendo en cuenta el nivel de medición marcado (los puestos de trabajo): la definición de la medida, las entradas y salidas (qué se mide) y los factores (qué influye).

En lo que respecta a la definición, es importante mencionar, tal y como Gummesson (1995) observó, que la calidad, la productividad y la rentabilidad forman una triplete en la que todas las partes están relacionadas con el mismo fenómeno: el resultado económico de la organización. En IS, la calidad de los productos puede ser significativa y estar afectada por el capital humano empleado, los factores del proceso de desarrollo software, y los recursos de despliegue en las etapas iniciales de desarrollo del producto, especialmente en el diseño (Krishnan, Kriebel, Kekre, & Mukhopadhyay, 2000). Esto se debe a que la industria IS, en gran medida, es un sistema abierto en el cual las partes interesadas, los clientes y los usuarios finales afectan a las entradas y salidas, producen una contribución a la eficiencia interna y externa, y por lo tanto a la productividad del proceso productivo (ver p. ej.: J. S.-C. Hsu, Chen, Jiang, & Klein, 2010). Por consiguiente, es necesario un enfoque totalmente diferente de la productividad para obtener una medida global que establezca cómo de bien utilizan las organizaciones los recursos para crear salidas con la calidad percibida necesaria y el valor requerido por el cliente (Grönroos & Ojasalo, 2004). Así pues, la definición en este tipo de industria debe tener en cuenta la calidad, además de la cantidad.

Con respecto a los factores que afectan a los resultados de la medición se puede señalar que son diversos, son dependientes del entorno y la organización (Wagner & Ruhe, 2008b), y por lo que deben ser analizados caso por caso (Paiva, et al., 2010). A nivel de puesto de trabajo, algunos de estos factores son: la motivación, el salario, el ambiente de trabajo, el compromiso organizacional, o la carrera profesional. Sin embargo, la literatura se centra principalmente en otros factores tales como la experiencia, el lenguaje de programación empleado, la consistencia de los requisitos, la reutilización de software y el tamaño del proyecto (Paiva, et al., 2010). Así pues, es necesario identificar, describir y medir, en la medida de lo posible, las características del entorno antes de realizar una medición de productividad.

Además de los factores, para elaborar un producto o dar un servicio (salida), es necesario un conjunto de recursos (entradas). Estas salidas y entradas pueden ser de naturaleza tangible o intangible; en los proyectos de desarrollo software coexisten ambos tipos de entradas y salidas, y es necesario tener en cuenta las características de ambos grupos. La medición de estos dos conjuntos de elementos, junto con los factores, sirve como base para medir la productividad.

En los trabajos del conocimiento, como es el caso de los proyectos de desarrollo de software, la principal, y casi única, entrada considerada es el tiempo empleado, en alguna medida como horas o días (Gómez, et al., 2008). En esta actividad, el esfuerzo en recursos humanos, es el principal coste por lo que el tiempo empleado en desarrollar el software es, como consecuencia, el principal recurso medible (Jørgensen & Shepperd, 2007). De forma complementaria, elementos tales como la motivación y la satisfacción laboral son entradas y salidas que deben tenerse en cuenta en la medición de la productividad en este nivel de medición.

Desde el punto de vista de otros recursos, el coste de los equipos y el hardware está continuamente descendiendo, y podría decirse de forma general que el coste de los mismos (actualmente) es casi irrelevante en proyectos de gran escala en comparación con el coste derivado del capital humano empleado. Así pues, el foco de atención sobre

las entradas a considerar para medir la productividad estará relacionado, de forma directa o indirecta, con el capital humano. En esta dirección, el tiempo es, sin lugar a duda, uno de ellos dado que a mayor tiempo empleado mayor es la cantidad de capital humano empleado. Por otro lado, la salida más ampliamente utilizada sigue la idea empleada con las entradas, es decir, se centra en la entrega del producto a tiempo. De este modo, se mide el tamaño del producto en SLOC o por su funcionalidad en PF (Albrecht & Gaffney, 1983). Sin embargo, estas dos salidas medidas no son las únicas que pueden ser medidas en IS (Petersen, 2011). Una salida importante, la cual está fuera de las habitualmente consideradas, es la calidad, la cual afecta a la salida producida (efectividad) y al proceso de producción en sí mismo (eficiencia) (Al-Darrab, 2000). Además, hay otras características de las salidas producidas que pueden afectar a los resultados de productividad, tanto por su creación inicial como por su utilización posterior.

Finalmente, es necesario tener en cuenta el nivel de medición en cualquier proceso de establecimiento y medición. En el caso de la productividad, puede ser definida desde varios niveles: sector, organización, unidad, proyecto, equipo de trabajo, e individual. En este caso, dado que el objetivo de investigación se centra en el puesto de trabajo, es decir, en el individual, la revisión se centrará en los niveles más próximos: organización, proyecto, y puesto de trabajo. Estos tres niveles han sido tratados con diferentes grados de interés por la comunidad científica previamente. En concreto, los estudios se centran principalmente en el nivel organizacional (ver p. ej.: Krishnan, et al., 2000), y en el nivel de proyecto (ver p. ej.: Maxwell, et al., 1996), mientras que el nivel de trabajador está poco investigado (Hernández-López, et al., 2013; Petersen, 2011). A nivel de organización, la productividad suele ser considerada desde el punto de vista económico, esto es, el beneficio de la organización, dado que en muchos casos se relacionan los conceptos (Grifell-Tatjé & Lovell, 1999). A este nivel también se tiene en cuenta el rendimiento (Tangen, 2005). A nivel de proyecto, la productividad suele estar definida por la eficiencia de recursos de los empleados y de los procesos de trabajo (Blackburn, Scudder, & Wassenhove, 1996). Y finalmente, a

nivel de trabajador, la productividad suele estar definida por el desempeño del trabajador (Rasch & Tosi, 1992). En esta investigación, el objetivo se centra en el nivel de puesto de trabajo y por lo tanto sólo se analizará de forma específica dicho nivel.

A continuación se presenta el estado de la cuestión referente a cada uno de estos temas de forma más precisa: (1) definición de productividad, (2) factores, (3) entradas y salidas, y finalmente (4) motivación y satisfacción laboral.

2.2.1. Definición de productividad

La definición de lo que se quiere medir es un elemento clave para cualquier medición dado que sin este elemento es imposible establecer una medida (Tangen, 2005). De forma inversa, es posible decir que una forma de conocer qué se quiere medir es obtener la definición de la medida (Sink, et al., 1984). A pesar de las diferentes definiciones de productividad, estas pueden ser clasificadas en tres grupos (Ghobadian & Husband, 1990): (1) *tecnológicas*: relación entre ratios de salidas y entradas utilizadas, (2) *ingenieriles*: la relación entre la salida actual y la potencial de un proceso, y (3) *económicas*: la eficiencia de la asignación de recursos. La productividad en IS es comúnmente medida utilizando una medida tecnológica basada en un ratio entre el tamaño del producto desarrollado y el esfuerzo requerido para producirlo (MacCormack, Kemerer, Cusumano, & Crandall, 2003), por ejemplo las líneas de código por unidad de tiempo (SLOC/t) (Maxwell, et al., 1996) o alguna variante de puntos función por unidad de tiempo (PF/t) (Low & Jeffery, 1990). En esta línea, la norma IEEE 1045-1992 define la productividad como la relación de una primitiva de salida (líneas de código, puntos función o documentos) y su correspondiente primitiva de entrada (esfuerzo, tiempo) para desarrollar software. Por otro lado, la norma ISO 9126-4 define la productividad basándose en factores de calidad como la capacidad del producto software para permitir a los usuarios emplear cantidades de recursos adecuados en relación con la efectividad alcanzada en un contexto de uso específico. Esta norma está centrada en la calidad, el usuario final y el

contexto de uso, por lo que la definición de la productividad gira en torno a estos tres conceptos. Además, esta norma, en su Anexo F, presenta tres medidas de productividad relacionadas con los tipos de recursos empleados: Productividad Humana (Efectividad/Esfuerzo), Productividad Temporal (Efectividad/Tiempo), y Productividad Económica (Efectividad/Coste). Las definiciones aportadas por ambas normas son complementarias, en cuanto a que la visión de la productividad es distinta (Cheikhi, Al-Qutaish, & Idri, 2012); la norma IEEE 1045-1992 se centra en las salidas y las entradas, y la norma ISO 9126-4 en la calidad, el usuario final y el contexto de uso.

A nivel lingüístico, la Real Academia de la Lengua Española, define *productividad* como:

1. Calidad de productivo.
2. Capacidad o grado de producción por unidad de trabajo, superficie de tierra cultivada, equipo industrial, etc.
3. Relación entre lo producido y los medios empleados, tales como la mano de obra, materiales, energía, etc.

Y *productivo* cómo,

1. Que tiene virtud de producir.
2. Que es útil o provechoso.
3. Que arroja un resultado favorable de valor entre precios y costes.

En estas definiciones están presentes tanto la visión de eficiencia (tercera acepción de *productividad*) de los recursos para producir productos y/o servicios, y la visión económica (tercera acepción de *productivo*) mencionadas por (Ghobadian & Husband, 1990). Así pues, considerando las definiciones existentes es posible decir que no existe una definición única de productividad y esta depende de la fuente consultada. No obstante, se puede decir que en las definiciones siempre existe un tipo de relación

entre los elementos utilizados (entradas), bien sean recursos o su transformación en costes, y los elementos producidos, tanto productos como servicios (salidas).

2.2.2. Factores

Los factores que afectan a la productividad han sido estudiados desde la concepción de la IS. Un ejemplo de esta vinculación son los factores (*drivers*) incluidos en el famoso modelo de estimación COCOMO (Boehm, 1981b; Boehm, Abts, Brown, et al., 2000). Por ejemplo, incluye elementos tales como la calificación de los analistas (ACAP), la experiencia en aplicaciones similares (AEXP), la calificación de los programadores (PCAP), la experiencia en el lenguaje de programación (LEXP), el uso de prácticas modernas de programación (MODP), entre otros. Otros factores que afectan a la productividad son: las restricciones de recursos, la volatilidad de los requisitos, la utilización de herramientas software, la complejidad del programa, la involucración del cliente y el usuario, la experiencia y capacidades del personal, la estabilidad del personal, el tamaño del equipo, el ratio de *outsourcing*... (Maxwell & Forselius, 2000; Premraj, Shepperd, Kitchenham, & Forselius, 2005). Algunos estudios se centran de forma específica en algunas tareas, por ejemplo, Anselmo y Ledgard (2003) indican algunos factores vinculados con el diseño y análisis: la independencia de los módulos, la comprensibilidad del código y la arquitectura, la flexibilidad del proceso de desarrollo software, la visibilidad de la arquitectura, la abstracción del proceso de desarrollo software para ser examinado. Dada la diversidad de factores, es interesante disponer de una clasificación de los mismos. Por ejemplo, Scacchi (1994) dividió estos factores en tres listas de atributos: (1) entorno de desarrollo software, (2) producto del sistema software, y (3) atributos del personal del proyecto. Además, las recientes revisiones sistemáticas de la literatura sobre este objeto de estudio (Paiva, et al., 2010; Sudhakar, Farooq, & Patnaik, 2011; Trendowicz & Münch, 2009; Wagner & Ruhe, 2008b) Su aportan una visión resumida de este elemento vinculado con la medición de la productividad. Ver la Tabla 3 que incluye un listado de algunos de los principales factores analizados en la literatura (se ha creado la tabla a partir de tres

recientes revisiones de la literatura sobre factores y la taxonomía de factores creada por el grupo GUFPI-ISMA en cuya traducción al español ha participado el autor de la presente tesis doctoral).

(Wagner & Ruhe, 2008a)	(Paiva, et al., 2010)	(Sudhakar, et al., 2011)	(Trendowicz & Münch, 2009)	(Santillo & Moretto, 2011)
Precedentes				
Confiabilidad del Software				
Tamaño de la Base de Datos			Tamaño de la Base de Datos y Complejidad	
Complejidad del Producto			Complejidad del Software Complejidad de Código Fuente	Complejidad del Producto
Restricciones de Tiempo de Ejecución			Restricciones del Proyecto	
Restricciones de Almacenamiento Principal			Restricciones del Proyecto	
Tamaño del Software				Tamaño del Proyecto
Calidad del Producto			Calidad de Software Requerida	
Interfaz de Usuario				
Flexibilidad del Desarrollo				

(Wagner & Ruhe, 2008a)	(Paiva, et al., 2010)	(Sudhakar, et al., 2011)	(Trendowicz & Münch, 2009)	(Santillo & Moretto, 2011)
Reutilización			Reutilización Grado de Reutilización Calidad de los Activos Reutilización	Reutilización Requisitos de Reutilización
Resolución del Riesgo de la Arquitectura				
Madurez del Proceso	Nivel de Madurez		Madurez de Proceso y Estabilidad	Madurez de la Organización
Volatilidad de la Plataforma				
Prototipado Temprano	Prototipado			
Complejidad del Diseño				
Duración del Proyecto	Tamaño del Proyecto			Tamaño del Proyecto
Hardware en Desarrollo Concurrente				
Uso de Herramientas Software	Herramientas de Desarrollo		Uso de Herramientas Herramientas Case Herramientas de Test	Herramientas de Desarrollo
Lenguaje de Programación	Lenguaje de Programación		Lenguaje de Programación	Lenguaje de Programación
Uso de Prácticas Modernas de Desarrollo	Metodología Modernidad		Uso de Prácticas Modernas de Desarrollo	Metodología

(Wagner & Ruhe, 2008a)	(Paiva, et al., 2010)	(Sudhakar, et al., 2011)	(Trendowicz & Münch, 2009)	(Santillo & Moretto, 2011)
Encaje de la Documentación con las necesidades del Ciclo de Vida	Documentación			Documentación Requerida
Credibilidad y Respeto				
Justicia				
Camaradería	Relaciones Interpersonales	Clima Laboral del Equipo		
Identidad de Equipo	Relaciones Interpersonales	Clima Laboral del Equipo		
Sentido de Élite				
Claridad de Objetivos				
Rotación			Rotación de Personal	Estabilidad del Equipo
Cohesión del Equipo	Relaciones Interpersonales	Clima Laboral del Equipo	Competencias de Trabajo en Equipo Cohesión del Equipo	Cohesión entre los Stakeholder
Comunicación	Relaciones Interpersonales		Comunicación entre el Equipo	
Apoyo a la Innovación				
Temperamento de los Desarrolladores		Características de los Miembros de Equipo		
Competencias de Analista		Características de los Miembros de Equipo	Competencias de Analista	Competencias del Personal
Competencias de Programación		Características de los Miembros de Equipo	Competencias de Programación	Competencias del Personal

(Wagner & Ruhe, 2008a)	(Paiva, et al., 2010)	(Sudhakar, et al., 2011)	(Trendowicz & Münch, 2009)	(Santillo & Moretto, 2011)
Experiencia en la Aplicación	Experiencia	Características de los Miembros de Equipo	Dominio Experiencia y Familiaridad en la Aplicación	Conocimiento del Dominio
Experiencia en la Plataforma	Dominio de la Aplicación Experiencia	Características de los Miembros de Equipo	Experiencia y Familiaridad en la Aplicación Plataforma Objetivo	
Experiencia con el Lenguaje y las Herramientas	Experiencia	Características de los Miembros de Equipo	Experiencia de Lenguaje de Programación Experiencia con las Herramientas	
Competencias del Gestor	Gestión del Proyecto	Comportamiento del Líder del Equipo	Experiencia y Habilidades del Gestor de Proyecto	Competencias de Gestión
Experiencia del Manager con la Aplicación	Experiencia	Características de los Miembros de Equipo	Experiencia y Habilidades del Gestor de Proyecto	Competencias de Gestión
Lugar de Trabajo Adecuado	Entorno de trabajo			
<i>E-factor</i>				
Fragmentación del Tiempo				
Separación Física	Teletrabajo		Desarrollo descentralizado/ multi-sitio	
Instalaciones de Telecomunicaciones				
	Acceso libre a Internet			
	Gestión del Conocimiento			

(Wagner & Ruhe, 2008a)	(Paiva, et al., 2010)	(Sudhakar, et al., 2011)	(Trendowicz & Münch, 2009)	(Santillo & Moretto, 2011)
	Motivación		Motivación del Equipo y Compromiso	
	Facilidad de Acceso al Trabajo			
	Tamaño del Equipo		Tamaño del Equipo	Tamaño del Equipo
	Salario			
	Brecha Tecnológica			
	Test		Testing	
	Formación		Nivel de Formación	
	Tipo de Proyecto (Desarrollo/Mantenimiento)		Tipo de Desarrollo	Tipo de Proyecto
	Estación de trabajo (PC)			
		Diversidad Del Equipo		
		Apoyo de la Alta Dirección		
		Conflictos dentro del Equipo		
			Complejidad de la Arquitectura	Arquitectura del Producto
			Complejidad de la Interfaz con otros Sistemas	Integración con Otros Sistemas
			Restricciones del Proyecto	
			Experiencia en tareas específicas	

(Wagner & Ruhe, 2008a)	(Paiva, et al., 2010)	(Sudhakar, et al., 2011)	(Trendowicz & Münch, 2009)	(Santillo & Moretto, 2011)
			Experiencia Total del Personal	
			Organización del Equipo	
			Estructura del Equipo	
			Revisiones e Inspecciones	
			Gestión de requisitos	Compleitud de Requisitos
			Estilo y Calidad de la Gestión	
			Complejidad de Interfaz Hardware y Software	
			Complejidad Lógica de los Problemas	
			Experiencia de Diseño	
			Experiencia Tecnología IT	Conocimiento de la Tecnología
			Uso de métodos	Metodología
			Presiones de Calendario	Restricciones Temporales
				Logística del Proyecto
				Integración con Otros Proyectos
				Entorno Tecnológico
				Cambios Tecnológicos

(Wagner & Ruhe, 2008a)	(Paiva, et al., 2010)	(Sudhakar, et al., 2011)	(Trendowicz & Münch, 2009)	(Santillo & Moretto, 2011)
				Restricciones Tecnológicas

Tabla 3. Factores Incidentes en la Productividad

A nivel de puesto de trabajo, los factores que afectan directamente a la productividad, siguiendo la clasificación presentada preliminar por Santillo y Moretto (2011) y posteriormente publicada por GUFPI-ISMA (2012)³ son los referentes al personal del proyecto junto con los relativos a las condiciones de trabajo definidas en el puesto de trabajo. No obstante, de forma indirecta los factores de proceso, producto, y tecnológico presentados por Santillo y Moretto deben ser considerados ya que en el caso de disponer de una correcta definición de puesto de trabajo, como la planteada en (Fernández-Ríos, 1995), el proceso corresponde al cómo lo hace, el producto a qué hace, y los tecnológicos al con qué lo hace. De este modo, la vinculación entre la definición de los puestos de trabajo y la medición de la productividad queda señalada, y puede indicarse que la definición del puesto de trabajo aporta la información de partida suficiente como para plantear una posible medida de productividad.

Los puestos de trabajo bajo estudio se caracterizan por ser puestos con un alto uso de capital humano por lo que los factores que más pueden influir en la productividad son los relativos al capital humano. De este modo, factores tales como la motivación, tratada inicialmente en los años 20 del siglo pasado por Elton Mayo, y en IS tenida en cuenta desde hace años (Beecham, Baddoo, Hall, Robinson, & Sharp, 2008; Boehm, 1981a) influyen en la productividad. Sin embargo, no está claro qué motiva a los ingenieros software, cómo están motivados, ni las salidas y beneficios de dicha motivación (Beecham, et al., 2008). Pese a ello, la noción general es disponer de

³ El autor de la presente tesis ha participado en la traducción de la taxonomía de factores al español, la cuál está pendiente de ser publicada.

trabajadores motivados como fuente de buenos resultados para los proyectos de desarrollo software (Hall, Sharp, Beecham, Baddoo, & Robinson, 2008). Además, existen factores, incluidos también en otras medidas como las relacionadas con la estimación, que influyen en la productividad; p. ej., la experiencia en las tareas (análisis, diseño, programación...) o en los lenguajes de programación empleados (Boehm, 1981b).

Desde el punto del autor de la tesis, los factores de personal requieren una especial atención, (ver p. ej.: Colomo-Palacios, Tovar-Caro, et al., 2010), máxime cuando el objeto de investigación son los puestos de trabajo. Las actividades de IS son intensivas en capital humano, por lo que el factor humano tiene que ser considerado en cualquier práctica de gestión relacionada con el mismo. En el contexto de la medición de la productividad, es conocido que factores relacionados con el personal tales como las capacidades técnicas y no técnicas, las habilidades, y la experiencia (en diversas áreas: programación, lenguajes de programación, diseño, análisis...) afectan directamente a la productividad pese a que la literatura que trate este asunto sea limitada (Pfleeger, 2008). Además de estos factores, y considerando la falta de literatura relacionada con esta área, cabe mencionar otros factores que pueden influir en los resultados de productividad, tales como la motivación (Beecham, et al., 2008), la prácticas vinculadas con los resultados y la evaluación dentro de la gestión del desempeño, los sistemas de compensación y beneficios, el clima organizacional y la felicidad de los trabajadores.

Pese a que existe un amplio número de estudios que tratan los factores que afectan a este tipo de puestos de trabajo, el signo y el grado con el que éstos influyen en la productividad, y cómo se afectan entre sí, es una tarea difícil pero necesaria que no ha sido realizada de forma concluyente hasta el momento (Wagner & Ruhe, 2008b); aunque existen algunos estudios que tratan de arrojar luz sobre este problema (ver p. ej.: Rodríguez, Sicilia, García, & Harrison, 2012). En esta dirección, se abre un amplio abanico de posibilidades de investigación al combinar el conocimiento de la gestión de recursos humanos con la gestión de la productividad dentro de la IS. De modo que

esta combinación, puede llevar a una transferencia de conocimiento con un propósito común de investigación (Koskinen, 2008).

2.2.3. Entradas y salidas

Para producir un producto o servicio (salida), es necesario un conjunto de recursos (entradas). Estas salidas y entradas pueden ser de naturaleza tangible o intangible. Por ejemplo, los recursos materiales empleados para producir un dispositivo electrónico son de naturaleza tangible mientras que el conocimiento necesario para diseñar el dispositivo es intangible. A su vez, el dispositivo electrónico es una salida de naturaleza tangible, mientras que la satisfacción del usuario con el dispositivo es intangible. En los proyectos de desarrollo software coexisten también ambos tipos de entradas y salidas, y es necesario tener en cuenta las características de ambos grupos. La medición de estos dos conjuntos de elementos, junto con los factores, sirve como base para medir la productividad.

En los trabajos del conocimiento, como es el caso de los proyectos desarrollo de software, la principal, y casi única, entrada habitualmente considerada es el tiempo empleado, en alguna medida como horas o días (Gómez, et al., 2008). El uso del tiempo, como única entrada, quizá se deba a dos hechos importantes para el éxito de los proyectos: por un lado, la entrega del proyecto en tiempo, y por otro lado, el tiempo de puesta en mercado del producto. Además, el esfuerzo en personal, es el principal coste en las organizaciones productoras de software, por lo que el tiempo empleado en desarrollar el software es, como consecuencia, el principal recurso medible (Jørgensen & Shepperd, 2007). El uso del tiempo, como única entrada encaja en la definición *economista* de productividad (Ghobadian & Husband, 1990), pero la productividad no es sólo un indicador económico. Además, el empleo del tiempo como única entrada plantea varias cuestiones no resueltas: ¿sólo se emplea el tiempo para producir software?; y si el tiempo es la única entrada medida, ¿qué tiempo debería ser medido, la cantidad de tiempo empleada por el personal para llevar a cabo sus tareas

o el tiempo pagado por el empleador? Desde el punto de vista de otros recursos, el coste de los equipos y el hardware es casi irrelevante, y podría decirse de forma general que el coste de los mismos (actualmente) es casi irrelevante en proyectos de gran escala en comparación con el coste derivado del capital humano empleado. Así pues, el foco de atención sobre las entradas a considerar para medir la productividad estará relacionado, de forma directa o indirecta, con el capital humano. En esta dirección, el tiempo es, sin lugar a duda, uno de ellos dado que a mayor tiempo empleado mayor es la cantidad de capital humano empleado, pero no parece ser la única entrada cuando IS ha llegado a ser considerada una actividad artesanal (Boehm & Ross, 1989).

Con respecto a las salidas, la más ampliamente utilizada sigue la idea empleada con las entradas, es decir, se centra en la entrega del producto a tiempo. De este modo, se mide el tamaño del producto en SLOC o por su funcionalidad en PF (Albrecht & Gaffney, 1983). Estas medidas de salidas están ampliamente aceptadas en las organizaciones a pesar de sus limitaciones. La funcionalidad se mide con PF (en alguna de sus variantes⁴) lo que representa la cantidad de funcionalidad desarrollada y, por consiguiente, es una medida de cuánto se entrega al cliente. El código fuente se mide en SLOC lo que representa el tamaño en líneas de código del producto desarrollado, lo cual es una medida de bajo nivel más tangible que los PF, pero que de igual forma mide cuanto se entrega. Quizá esta focalización en el tamaño se deba a estudios tales como (Agrawal & Chari, 2007) en los que se señala que el único parámetro que afecta a la productividad, y la calidad, es el tamaño del producto. Sin

⁴ Algunas de estas variantes son: ISO/IEC 24570:2005 NESMA Guide to Using Function Point Analysis, ISO/IEC 20968:2002 Mk II Function Point Analysis - Counting Practices Manual, ISO/IEC 19761:2011 COSMIC-FFP - A Functional Size Measurement Method, ISO/IEC 20926:2009 IFPUG 4.3.1 Unadjusted functional size measurement method - Counting practices manual, UNE 71045-1:2000. "Tecnología de la información. Medida del Software. Medida del tamaño funcional. Parte 1: Definición de conceptos."

embargo, estas dos salidas no son las únicas dos que pueden ser medidas en IS (Petersen, 2011). De Aquino y De Lemos Meira han identificado cuatro categorías de salidas (de Aquino Junior & De Lemos Meira, 2009): *tamaño físico* (p. ej., SLOC), *tamaño de diseño* (p. ej., número de módulos), *tamaño funcional* (p. ej., PF), y valores basados en *métricas de valor* o que usan modelos multidimensionales para evaluar los diferentes aspectos de lo que se produce en un proyecto software. Así pues, aunque hay otras alternativas al tamaño físico, de diseño, o funcional, se continúan empleando principalmente las medidas de salidas centradas en el tamaño, dejando fuera otras salidas. Por ejemplo, una salida importante, la cual está fuera de las habitualmente consideradas, es la calidad, la cual afecta a la salida producida (efectividad) y al proceso de producción en sí mismo (eficiencia) (Al-Darrab, 2000). Además, hay otras características de las salidas producidas que pueden afectar a los resultados de productividad, tanto por su creación inicial como por su utilización posterior; ejemplos de estas salidas son la reutilización (Anselmo & Ledgard, 2003) y la documentación (Boehm, 1987).

2.2.3.1. Motivación y satisfacción laboral

La satisfacción laboral es una área de investigación antiguas dentro de la psicología industrial (Locke, 1969; Maslow, 1943; Vroom, 1964). El vínculo entre la satisfacción y el rendimiento/desempeño es casi tan antiguo como los propios conceptos, pero actualmente no se considera un área de investigación tan importante (Judge, Thoresen, Bono, & Patton, 2001). La satisfacción laboral ha sido estudiada en relación con múltiples conceptos importantes en la gestión de recursos humanos como la rotación (Mobley, 1977), la personalidad (Judge, Heller, & Mount, 2002), y el cumplimiento del trabajo (Morrow & McElroy, 1987) entre otros. A su vez, la motivación también ha sido ampliamente estudiada, p. ej. en relación con la carrera profesional (London, 1983), el *burnout* (Houkesa, Janssen, de Jongeb, & Nijhuis), el nivel salarial (Eisenberger, Rhoades, & Cameron, 1999), y en relación con el porqué los humanos están motivados para trabajar (Herzberg, Mausner, & Snyderman, 1959).

Dentro de IS estas áreas de investigación no son tan antiguas, dada su menor antigüedad en comparación con la investigación en el área industrial, pero ha sido abordado en diversas investigaciones, tal y como se concluye en una reciente revisión sistemática de la motivación en IS (Beecham, et al., 2008).

2.2.3.1.1. Motivación

La motivación ha sido investigada desde principios del siglo XX. Pero, de acuerdo con Golembiewski (2000), es posible encontrar más de 140 definiciones formales del término. A pesar de la variedad, estas definiciones tienen una serie de características comunes: la motivación es interna al individuo, tiene intensidad, fuerza y duración, varía en función del objetivo, y determina parte del comportamiento humano. En un primer momento, Taylor (1911) propuso en su teoría de gestión científica, o Taylorismo, que los humanos están sólo motivados (para trabajar) por el salario. En aquellos años, los trabajadores cobraban más si producían más por hora, lo que produjo un rápido aumento de la productividad pero los trabajadores caían exhaustos y empezaban a estar insatisfechos. Estas salidas psicológicas fueron posteriormente consideradas por dos de los principales trabajos sobre motivación y satisfacción laboral: la pirámide (o jerarquía) de necesidades de Maslow (1954) y la teoría de factor dual de Herzberg (1959). Herzberg introdujo la teoría dual (Herzberg, et al., 1959), también llamada de factores motivadores e higiénicos, la cual señala que existen dos conjuntos de necesidades: las necesidades del ser humano como animales para evitar el dolor, y las necesidades como humanos para crecer psicológicamente. Esta teoría complementa la teoría de la pirámide de necesidades de Maslow creada unos años antes (1943) la cual establece una jerarquía de necesidades con cinco niveles, desde necesidades básicas a complejas: (1) fisiológicas (comida, agua, dormir...), (2) seguridad (sentirse seguro, empleo, recursos, salud...), (3) amor/pertenencia (amistad, familia...), (4) autoestima (confianza, respeto por los demás, respetado por los demás...), (5) auto-realización (moralidad, creatividad...).

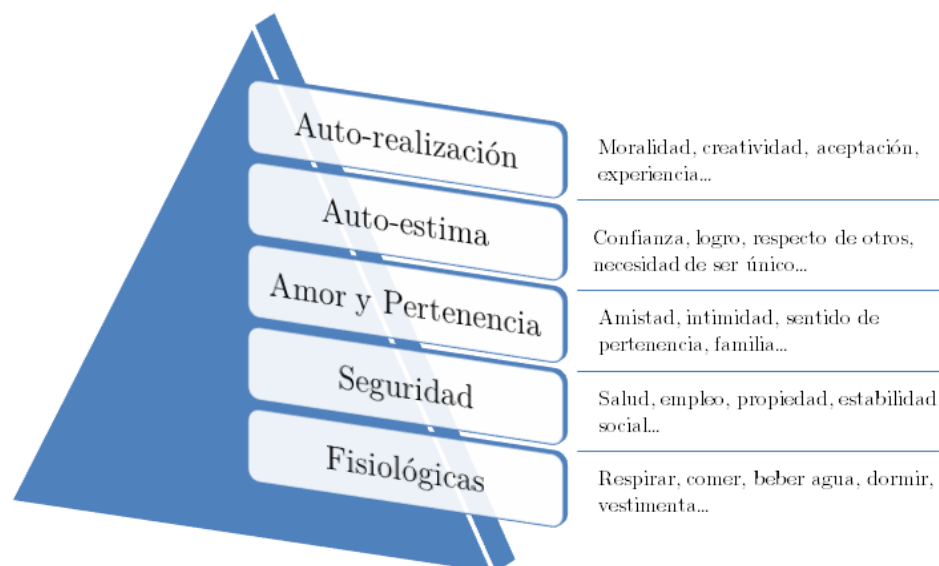


Ilustración 6. Pirámide de necesidades de Maslow (1943)

Desde el nivel uno al nivel tres de la pirámide de Maslow pueden considerarse como factores higiénicos en la teoría de Herzberg, y desde el nivel tres al nivel cinco como motivadores. Hay que tener en cuenta que si existen factores higiénicos sin cubrir producen desmotivación, y por lo tanto no es posible alcanzar la motivación con motivadores *reales*. De ahí el nombre de higiénicos ya que su presencia no influye positivamente en la motivación, sino que la reducen y deben ser cubiertos. Adicionalmente, las necesidades hasta el nivel cinco pueden ser clasificadas en un grupo denominado necesidades de déficit (*Defecit needs, D-needs*) las cuales son necesidades primordiales, y el nivel cinco como necesidades de ser (*Being needs, B-needs*). La diferencia estriba en que mientras que las necesidades de déficit pueden ser alcanzadas, las necesidades de ser son una fuerza impulsora continua. Y, tal y como indicó Maslow, una vez que una persona cubre las necesidades de un nivel específico, desarrollara necesidades superiores.

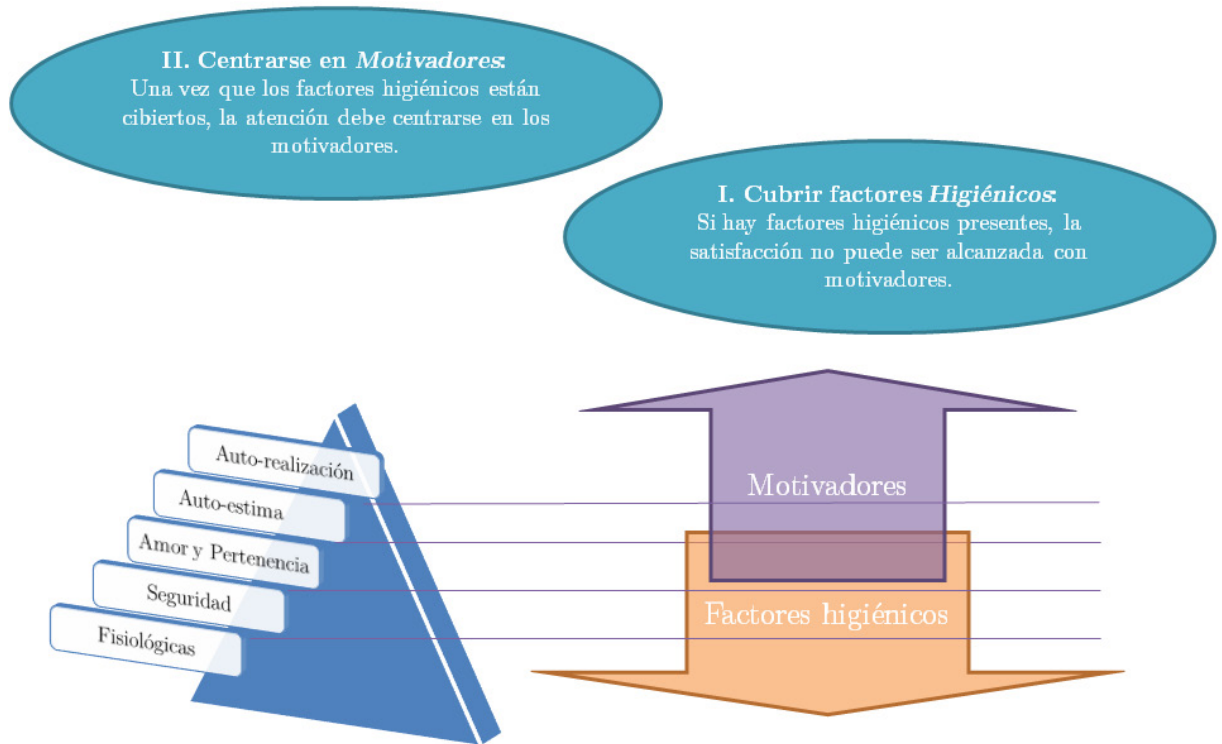


Ilustración 7. Teoría dual de Herzberg (1959)

Otra clasificación de los factores motivadores es la división en dos categorías: motivadores intrínsecos y motivadores extrínsecos (Herzberg, et al., 1959). Dentro de la primera categoría están factores como el reconocimiento, la aceptación social, la carrera profesional. Esta categoría de motivadores está dirigida por el interés o disfrute por la tarea por sí misma, y existe dentro del individuo en lugar de por presión externa, es decir, la motivación intrínseca está basada en la satisfacción de la actividad en lugar del obsequio conseguido por realizar la tarea. Por otro lado, la motivación extrínseca se relaciona con la recompensa por realizar una determinada actividad, lo que se contrapone con la motivación intrínseca. No obstante, tanto los factores intrínsecos y los extrínsecos, ayudan a facilitar el sentimiento de satisfacción en los trabajadores sobre su trabajo.

2.2.3.1.2. *Satisfacción laboral*

La satisfacción laboral también ha sido ampliamente tratada en la literatura de investigación. En sus orígenes fue un concepto un poco confuso, hasta que Locke (1969) mezcló algunos conceptos relacionados. Posteriormente introdujo la teoría de la

fijación de metas (*Range of Affect Theory*) (Locke, 1976) en la cual establece que la satisfacción está determinada por la discrepancia entre lo que uno quiere de un trabajo y lo que uno tiene en un trabajo. Uno de los factores de satisfacción laboral más tratados es el salario, pero su relación con la satisfacción continua siendo controvertida (Judge, Piccolo, Podsakoff, & Shaw, 2010). A su vez, otros factores tales como la satisfacción con el puesto de trabajo, es decir, la satisfacción con lo que el empleado hace (Judge & Church, 2000) o el entorno de trabajo en el que el empleado trabaja (Berry, 1997) pueden ser más importantes que el salario. Otros factores relacionados con la satisfacción laboral son las expectativas, el reconocimiento, y el crecimiento personal y profesional entre otros. Así pues, se puede decir que la satisfacción laboral es circunstancial y subjetiva para cada empleado y depende de cada situación y puesto.

Al igual que en el caso de la motivación, la importancia de la satisfacción laboral se encuentra ampliamente documentada pero continua habiendo ciertas controversias. Por ejemplo, la relación entre la satisfacción laboral y el desempeño laboral es una relación positiva pero el grado de influencia parece no ser universal (Judge, et al., 2001). Además, hay que tener en cuenta que la relación entre la satisfacción laboral y el desempeño es más intensa para trabajos con mayor dificultad que en otros de menor dificultad (Saari & Judge, 2004). Este es el caso de los puestos de trabajo en IS. Otros conceptos relacionados con la satisfacción laboral son el absentismo (Johns, 1997) y el abandono laboral (Carsten & Spector, 1987), que afectan en gran medida a la productividad en IS.

2.2.3.1.3. *Motivación y satisfacción laboral en IS*

En IS, la satisfacción laboral y motivación también han sido tratadas. Por un lado, la satisfacción laboral ha sido estudiada desde el punto de vista de las comunicaciones en el equipo (Javed, Manzil-e-Maqsood, & Durrani, 2004), del uso de metodologías ágiles frente a no ágiles (Melnik & Maurer, 2006), la retención del talento y la rotación de personal (Hall, Beecham, Verner, & Wilson, 2008; Westlund & Hannon, 2008), la

personalidad y las características de los procesos y tareas (Acuña, Gómez, & Juristo, 2009), entre otras. Por otro lado, la motivación ha sido estudiada también desde diversos puntos de vista, p. ej., del software libre (Ye & Kishida, 2003), la cultura (Couger, Adelsberger, Borovits, Zviran, & Motiwalla, 1990; Igbaria, Meredith, & Smith, 1994), la calidad y el éxito del proyecto (Procaccino, Verner, Shelfer, & Gefen, 2005), y ha sido señalado como uno el mayor modificador en la productividad de los profesionales en IS (Boehm, 1981b). Para resumir el estado de la cuestión sobre la motivación, una revisión sistemática de la literatura ha sido realizada recientemente (Beecham, et al., 2008). En esta revisión, los motivadores y desmotivadores para ser más o menos productivo fueron analizados. El motivador más citado es: "la necesidad de identificarse con la tarea" dentro del cual, tener objetivos claros, un interés personal, comprender el propósito de la tarea, y conocer cómo encaja la tarea en el proyecto son fuentes de motivación. Además, la importancia de la satisfacción laboral, el trabajar en una parte de un producto con calidad, la variedad de tareas, y la carrera profesional, también aparecen en esta revisión sistemática como motivadores.

Por otro lado, la motivación está relacionada con el desempeño y la productividad de los profesionales de IS. El desempeño individual, concepto más general que la productividad, puede ser definido como: la combinación de motivación, aptitudes y experiencia" (Curtis, et al., 1988). Y desde el punto de vista de la productividad, la motivación es un factor que continua influenciando fuertemente a la productividad de los profesionales de IS (Paiva, et al., 2010; Trendowicz & Münch, 2009; Yilmaz & O'Connor, 2012) pese al paso del tiempo desde las primeras vinculaciones entre estos elementos (Boehm, 1981b). Además, existen otros factores que relacionan ambos conceptos (productividad y motivación). En una revisión sistemática llevada a cabo por Wagner y Ruhe (2008b) sobre los factores que afectan a la productividad en IS, aparecen conceptos tales como el respecto, la justicia, la credibilidad, la claridad de los objetivos, la cohesión, las capacidades y habilidades, el lugar de trabajo, entre otros, que encajan en la categoría de motivadores de Herzberg (1959). Estos y otros factores aparecen en otros estudios recientes sobre los factores que influyen la productividad

(de Barros Sampaio, Barros, de Aquino Junior, e Silva, & de Lemos Meira, 2010; Melo, Cruzes, Kon, & Conradi, 2011; Paiva, et al., 2010; Trendowicz & Münch, 2009). Así pues, la importancia de la motivación y la satisfacción laboral está ampliamente analizada en la literatura general y en la específica de IS. Cabe mencionar, que si se considera como trabajadores del conocimiento (Drucker, 1999) a los ingenieros de software, la influencia de la motivación y la satisfacción laboral en estos profesionales deben ser tenida en cuenta en cualquier estudio relacionado con la productividad si se tiene en cuenta que es uno de los factores que más influyen la productividad a nivel de equipo e individual (Beecham, et al., 2008). Por ello, se incluyeron en esta fase de la investigación.

Finalmente, tal y como Roznowski y Hulin (1992) comentaron, parece asumirse que todos conocemos lo que hay que saber sobre estos conceptos, y por ello se pierde de vista su utilidad dada la familiaridad con los conceptos y su pasada popularidad. Por ello, la investigación en estas áreas necesita continuar (Beecham, et al., 2008; Sach, Sharp, & Petre, 2011). Además, es habitual incluir una parte relativa a satisfacción laboral en los estudios relativos a recursos humanos para complementar los resultados obtenidos dado que afecta a múltiples elementos. Por ello se incluye en el estudio del estado de la cuestión un apartado para tratar la relación entre estos dos conceptos y la productividad, y se incluirá en la fase cualitativa del enfoque del problema.

2.3. Cuestiones

Antes de considerar los nuevos retos en la medición de la productividad en IS, es necesario establecer algunos requisitos generales que guíen la formulación del concepto de productividad en el sector. Para ello, siguiendo los seis requisitos generales establecidos por Adam, Johanson, y Gravesen (1995), la medición de la productividad en IS requiere la presencia de las siguientes condiciones:

- La salida de IS tiene que ser vista como un valor para el cliente y desde la perspectiva del cliente.
- Las salidas de IS tienen que ser definidas por su nivel de calidad.
- El consumidor tiene que ser parte del concepto de productividad.
- Las medidas de productividad tienen que estar más relacionadas con el cliente.
- Los indicadores dinámicos de productividad tienen que ser utilizados en lugar de medidas estáticas de salida/entrada.
- Medidas para situaciones específicas tienen que estar disponibles para permitir la complejidad y la diversidad de las operaciones de la IS.

La industria IS es, en gran medida, un sistema abierto en el cual las partes interesadas (principalmente los clientes y los usuarios finales) afectan a las entradas y salidas, producen una contribución a la eficiencia interna y externa, y por lo tanto a la productividad del proceso productivo (ver p. ej.: J. S.-C. Hsu, et al., 2010). Por consiguiente, es necesario un enfoque totalmente diferente de la productividad para obtener una medida global que establezca como de bien utilizan las organizaciones IS los recursos para crear salidas con la calidad percibida necesaria y el valor para el cliente requerido (Grönroos & Ojasalo, 2004). Por lo tanto, las medidas de entradas y salidas deben tener en cuenta la cantidad y calidad. Además, las fronteras entre las entradas y las salidas son difusas (Gupta, 1995). Así pues, debería tenerse en cuenta la realización de un análisis profundo de las entradas ya que el uso de medidas *proxies* (medidas indirectas) introduce sesgos en la medición de la productividad. Además, esta fuente de error es una de las más difíciles de superar, y los posibles métodos para tratar con ello no son todos obvios (Rees, 1980).

Con respecto a la medición, definir “una unidad de ingeniería del software” es una tarea difícil. Así pues, hay algunas medidas universales de entradas y salidas para realizar medidas (Boehm, Abts, & Chulani, 2000). Desde el punto de vista de las entradas, algunas medidas de productividad utilizan las horas-hombre como una medida de esfuerzo, y las capacidades y competencias técnicas y no técnicas del personal como un factor correctivo para el esfuerzo. Desde el punto de vista de las

salidas, algunas medidas de productividad utilizan las SLOC como medida de cuantificación de la salida y el ratio de errores como un factor correctivo. Otras medidas ampliamente utilizada son los PF completados y los productos terminados. Estos parámetros son medidas físicas, que pueden ser transformados en unidades monetarias para obtener medidas financieras de productividad.

Una vez que las entradas, salidas, y factores que afectan a la medición de la productividad han sido definidos, es posible formular una medida de productividad. Con el objetivo de resumir el estado de la cuestión sobre entradas, salidas, y medidas, se presentan en la Tabla 4 las más representativas de cada categoría; en ella se encuentran ordenadas de acuerdo a la dificultad de medir, desde las más fáciles a las más difíciles. Desde el punto de vista del autor del proyecto, es sorprendente que a pesar de la controversia existente sobre la validez de algunas medidas de entradas y salidas para medir la productividad en IS se continúen utilizando. Ejemplo de ello son las medidas derivadas de la utilización de las SLOC como salida y de las Horas-Hombre como entrada (Boehm, 1987), o la utilización de los PF como salida, dada su alta correlación con las SLOC (Banker, Datar, & Kemerer, 1991; Laranjeira, 1990). Estas medidas continúan en uso en la actualidad en numerosas organizaciones pese a dicha controversia; probablemente bajo la ley que DeMarco y Lister (1999, p. 59) versionan de los principios de la capacidad de ser medido de Gilb: *"Todo lo que se necesita cuantificar puede ser medido de alguna forma mejor que no medirlo de ninguna forma"*⁵.

Entradas	Salidas	Medidas (P = Productividad)
Salarios	Ventas	$P = \text{Ventas} / \text{Salarios}$

⁵ Traducción de: *"Anything you need to quantify can be measured in some way that is superior to not measuring it at all."*

Entradas	Salidas	Medidas ($P = \text{Productividad}$)
Esfuerzo Horas Hombre	TLOC SLOC + DLOC	$P = \text{TLOC} / \text{Esfuerzo}$ (ver p. ej.: Jones, 1981; López-Martín, Chavoya-Peña, & Meda-Campaña, 2012; MacCormack, et al., 2003; Maxwell, et al., 1996; Moazeni, Link, & Boehm, 2013; Sison, 2009; Tan et al., 2009; Tausworthe, 1982; W. D. Yu, Smith, & Huang, 1991)
Esfuerzo Horas Hombre	PF (en cualquiera de sus variantes)	$P = \text{PF} / \text{Esfuerzo}$ (ver p. ej.: Bok & Raman, 2000; de Souza Carvalho, Rosa, dos Santos Soares, Teixeira da Cunha Junior, & Buiatte, 2011; Desharnais & April, 2010; Desharnais, Yıldızoğlu, April, & Abran, 2013)
Esfuerzo Horas Hombre	Instrucciones de código entregadas (DSI)	$P = \text{DSI} / \text{Esfuerzo}$ (ver p. ej., Gaffney, 1989)
Cualquiera	Cualquiera	Data Envelopment Analysis (DEA) (ver p. ej.: Asmild, et al., 2006; Banker, et al., 1991; Liping, Qiusong, Sun, Tong, & Wang, 2005; Mahmood, Pettingell, & Shaskevich, 1996; Ruan et al., 2007; Stensrud & Myrtveit, 2003; Yang & Paradi, 2004)
Cualquiera	Cualquiera	Medidas multifactor (ver p.ej., Kitchenham & Mendes, 2004)
Cualquiera	Cualquiera	Medidas de modelos lineales generales.

Tabla 4. Entradas y salidas utilizadas para medir la productividad

Finalmente, la medición de la productividad está relacionada con las prácticas de gestión, y más en concreto con la cultura organizacional y su estructura. En organizaciones en las cuales la cultura de informar y comparar está implantada, las medidas de productividad pueden ser consideradas como un posible indicador útil para medir el efecto de cualquier mejora en la productividad. Además, en organizaciones cuya gestión esté basada en estilo de mando directivo, donde las ordenes vienen de arriba hacia abajo, las medidas de productividad serán aceptadas como una forma de comunicar a los niveles superiores como de bien se están ejecutando los proyectos. Por otro lado, en organizaciones con una cultura de innovación y creatividad, las medidas de productividad podrían ser vistas como una herramienta de control que no sirve para la consecución de los objetivos. Adicionalmente, la unión entre los objetivos organizacionales y la productividad podría no estar claro en estas organizaciones, y

esto puede llevar a una falta de comprensión sobre el qué, el cómo y porqué es importante evaluar, y por tanto, medir la productividad.

2.4. Puestos de trabajo

Dado que es necesario considerar el nivel de medición, en este caso a nivel de puesto de trabajo, las entradas a considerar deberían ser las necesarias, para producir las salidas de dicho puesto. Esto puede obtenerse de la definición del puesto de trabajo. La estructura de cualquier organización, a pesar de su forma, requiere la definición de los puestos de trabajo que conforman su estructura (Fernández-Ríos, 1995). Estas definiciones son la base que debería emplearse para alcanzar los objetivos y misión de la organización. En concreto, una definición del puesto puede contener la siguiente (y otra) información: su rol y misión, su situación en el organigrama, las tareas que deben ser desempeñadas en el (incluyendo información sobre qué se necesita, cómo se realiza, para qué se realiza, la frecuencia, el tiempo dedicado, la autonomía, las relaciones con otros trabajos...), diagramas de flujo de trabajo, esfuerzo (intelectual y físico), riesgos, condiciones de trabajo, supervisión, conocimientos y habilidades necesarios, valores, y acceso a otros puestos dentro del plan de carrera de la organización (Hackman & Oldham, 1980). Además, hay que destacar que una misma persona puede desempeñar más de un puesto de trabajo en una organización, y que varias personas pueden desempeñar el mismo puesto. Así pues, la definición de los puestos de trabajo, junto con su valoración, es una necesidad para cualquier organización (Schuler & Jackson, 1987).

En IS, hay varios puestos de trabajo más o menos reconocidos universalmente, a pesar de que su definición varía en cada organización y están en constante actualización (Colomo-Palacios, Tovar-Caro, et al., 2010). Por ejemplo, los puestos de jefe de proyecto, programador o analista son puestos de trabajo referenciados tanto en la investigación como en la industria desde hace años pero su definición es difusa (Litecky, Aken, Ahmad, & Nelson, 2010; Yilmaz, O'Connor, & Clarke, 2012). A pesar

de la variedad de puestos de trabajo, en IS todos ellos pueden ser catalogados como de cuello blanco y/o trabajadores del conocimiento (Drucker, 1999; Ramirez & Nembhard, 2004). Estos puestos de trabajo están caracterizados por el uso de capital humano, intelectual y relacional para desempeñar las tareas y alcanzar los objetivos (Acuña, et al., 2006; Colomo-Palacios, Cabezas-Isla, García-Crespo, & Soto-Acosta, 2010). Estas características hacen que estos puestos sean diferentes a otras industrias en las cuales los trabajadores emplean principalmente la mano de obra desde un punto de vista físico, en lugar de intelectual. Además, los recursos que se necesitan en los puestos de trabajo en IS son principalmente intangibles (p. ej., conocimiento y experiencia) frente a los activos tangibles utilizados en las industrias tradicionales (Rus & Lindvall, 2002). Finalmente, el entorno de trabajo en el que las personas desempeñan sus tareas afecta tanto a la productividad a niveles superiores como a la individual. En IS, mediante la combinación de espacios de trabajo individuales para una mejor concentración en tareas técnicas para evitar distracciones y lugares comunes para una mejor comunicación, se produce un aumento de la productividad (Sommerville, 2010).

Así pues, las entradas utilizadas y las salidas producidas dependen de cada puesto de trabajo y deberían estar explícitas en la definición de cada uno de ellos (Fernández-Ríos, 1995; Hackman & Oldham, 1980). Por ejemplo, si consideramos el puesto de Analista bajo el punto de vista de Métrica 3 (2000), obtenemos que su responsabilidad es: *"[...] elaborar un catálogo detallado de requisitos que permita describir con precisión el sistema de información, para lo cual mantendrán entrevistas y sesiones de trabajo con los responsables de la organización y usuarios, actuando de interlocutor entre éstos y el equipo de proyecto en lo que a requerimientos se refiere. Estos requisitos permiten a los analistas elaborar los distintos modelos que sirven de base para el diseño, obteniendo los modelos de datos y de procesos en el caso del análisis estructurado y los modelos de clases e interacción de objetos en análisis orientado a objeto. Así mismo realizan la especificación de las interfaces entre el sistema y el usuario."* Con esta definición, puede decirse que las salidas producidas por un analista

son, entre otras, catálogos de requisitos, modelos de datos, modelos de procesos, modelos de clases e interacción, y especificaciones de interfaces. Con respecto a las entradas que un analista usa, puede decirse que utiliza a los usuarios, el conocimiento funcional del cliente, y el catálogo de requisitos. Además, para generar las salidas existe una interacción con otros puestos y con personas fuera de la organización por lo que estas interacciones podrían considerarse como otras entradas. Este ejemplo muestra la dificultad de medir la productividad a nivel de puesto de trabajo dado que ninguna de las entradas y salidas utilizadas habitualmente para medir la productividad están mencionadas de forma explícita, y se mencionan otras que no se tienen en cuenta habitualmente. Otra característica de este ejemplo es que muestra cómo la frontera entre las entradas y salidas es difusa: el catálogo de requisitos es generado por el analista pero también sirve como entrada para la construcción de los modelos; esto es un problema clásico de las industrias de servicios (Gupta, 1995). Otro ejemplo de estas fronteras difusas sucede durante la fase de mantenimiento, en la que todos los productos de los procesos de producción de software son utilizados como entradas y son transformados para adaptarlos a las necesidades del cliente, o para solucionar los *bugs* (errores) encontrados. Finalmente, otro factor a tener en cuenta es la calidad, tanto de las entradas utilizadas como de las salidas generadas (Procaccino, et al., 2005). Así pues, cabe pensar que las entradas y las salidas utilizadas en la medición de la productividad en IS deberían estar vinculadas con el puesto de forma específica, y no con una salida final del proceso de desarrollo de software y una entrada común a todos ellos. Por ello, tal y como se mencionó anteriormente, para construir nuevas medidas de productividad en IS es necesario identificar las entradas y salidas que podrían ser incluidas en dichas medidas.

2.4.1. Puestos de trabajo en proyectos de desarrollo de software

En los proyectos de desarrollo de software existen varios puestos de trabajo más o menos reconocidos universalmente, aunque su definición varía en cada organización y están en constante actualización (Garcia-Crespo, Colomo-Palacios, Gomez-Berbis, & Tovar-Caro, 2008). Por ejemplo, puestos tales como jefe de proyecto, programador o analista son puestos referenciados tanto en la academia como en la industria desde hace años pero su definición es difusa y no es universal (Litecky, et al., 2010; Yilmaz, et al., 2012). Además, es frecuente encontrar definiciones de los puestos de trabajo en función de las competencias necesarias para poder desempeñar dicho puesto, lo que produce una nueva perspectiva del diseño de puestos de trabajo tradicional (Colomo-Palacios, Cabezas-Isla, et al., 2010; Colomo-Palacios, Tovar-Caro, et al., 2010). No obstante, la definición de puestos de trabajo es compatible con la utilización y definición de competencias.

Una fuente de consulta de puestos de trabajo es el proyecto O*Net creado por el departamento de trabajo de los EE.UU (<http://www.onetonline.org/>). En él es posible consultar descripciones genéricas de puestos de trabajo muy útiles a la hora de crear nuevas o actualizar definiciones de puestos de trabajo existentes. En O*Net se definen algunos de los puestos más comunes en IS de forma genérica. Entre ellos se encuentran:

- **Computer Programmer.** Este puesto de trabajo tiene por objetivos crear, modificar y probar el código, los formularios, y scripts que permiten a las aplicaciones ejecutarse. Para ello, parten de las especificaciones. Pueden ayudar a los *Software Developers* a analizar las necesidades del cliente y diseñar soluciones software. Además, pueden desarrollar y escribir soluciones para almacenar, localizar, y recuperar documentos específicos, datos, e información.

- La definición de este puesto de trabajo también ha sido mencionada como *Programmer Analyst, Programmer, Analyst Programmer, Computer Programmer, Software Developer, Applications Developer, Computer Programmer Analyst, Internet Programmer, Java Developer, Web Programmer*.
- Para más información consultar
<http://www.onetonline.org/link/summary/15-1132.00>
- **Software Developer (Systems Software).** Este puesto de trabajo tiene por objetivos investigar, diseñar, desarrollar, y probar software y aplicaciones informáticas en general. Además, tiene que establecer especificaciones operativas y formular y analizar los requisitos software. Puede diseñar sistemas software embebidos. Para ello aplica principios y técnicas de ingeniería informática, ingeniería del software y análisis matemático.
 - La definición de este puesto de trabajo también ha sido mencionada como *Software Engineer, Software Developer, Systems Engineer, Network Engineer, Developer, Publishing Systems Analyst, Application Developer, Averaged HEND (High-Energy Neutron Data) Data Product Lead (AHD Product Lead), Business Systems Analyst, Computer Consultant*.
 - Para más información consultar
<http://www.onetonline.org/link/summary/15-1133.00>
- **Software Developer (Applications).** Este puesto de trabajo tiene por objetivos desarrollar, crear, y modificar aplicaciones informáticas generales o programas específicos. Además, tiene que analizar las necesidades de los usuarios y desarrollar soluciones software, diseñar y ajustar el software para el uso del cliente con el propósito de optimizar la eficiencia operacional. Además, puede analizar y diseñar bases de datos dentro del área de la aplicación, trabajar individualmente o coordinando el desarrollo de la bases

de datos como parte de un equipo. Puede a su vez supervisar *Computer Programmers*.

- La definición de este puesto de trabajo también ha sido mencionada como *Software Engineer, Application Integration Engineer, Programmer Analyst, Software Development Engineer, Computer Consultant, Software Architect, Software Developer, Technical Consultant, Applications Developer, Business Systems Analyst*.

- Para más información consultar

<http://www.onetonline.org/link/summary/15-1132.00>

- **Computer Systems Analyst.** Este puesto de trabajo tiene por objetivos analizar los problemas de procesamiento de datos científicos, ingenieriles, de negocio, entre otros para implementar y mejorar los sistemas informáticos. Además, analiza los requisitos de los usuarios, los procedimientos, y los problemas para automatizar y mejorar los sistemas actuales y revisa las capacidades de los sistemas informáticos, los flujos, y las limitaciones de horarios. Y puede analizar y recomendar soluciones comerciales existentes.

- La definición de este puesto de trabajo también ha sido mencionada como *Systems Analyst, Programmer Analyst, Business Systems Analyst, Computer Systems Analyst, Computer Systems Consultant, Computer Analyst, Information Systems Analyst (ISA), Applications Analyst, Business Analyst, Systems Engineer*.

- Para más información consultar

<http://www.onetonline.org/link/summary/15-1121.00>

- **Computer and Information Systems Manager.** Este puesto de trabajo tiene por objetivos planificar, dirigir, y coordinar las actividades y proyectos en campos tales como el procesamiento de información electrónica, los sistemas de la información, análisis de sistemas, y programación de software.

- La definición de este puesto de trabajo también ha sido mencionada como *Information Technology Manager (IT Manager), Information*

Technology Director (IT Director), Information Systems Director (IS Director), Data Processing Manager, MIS Director (Management Information Systems Director), Information Systems Manager (IS Manager), Information Systems Supervisor (IS Supervisor), Computing Services Director, Director of Application Development, Technical Services Manager.

- Para más información consultar:

<http://www.onetonline.org/link/summary/11-3021.00>

- **Software Quality Assurance Engineers and Testers.** Este puesto de trabajo tiene por objetivos desarrollar y ejecutar planes de prueba de software para identificar los problemas software y sus causas.

- La definición de este puesto de trabajo también ha sido mencionada como *Quality Assurance Analyst (QA Analyst), Quality Assurance Director (QA Director), Software Quality Assurance Engineer (SQA Engineer), Software Quality Engineer, Product Assurance Engineer, Software Test Engineer.*

- Para más información consultar

<http://www.onetonline.org/link/summary/15-1199.01>

Finalmente, y a diferencia de otros sectores, los puestos de trabajo en IS están muy influenciados por la forma en la que el trabajo se estructura. En la mayoría de las organizaciones IS, se ha implantado la organización del trabajo en equipos de trabajo (Carmel, 1999; Gorla & Lam, 2004; Sawyer, 2004), lo que está contrapuesto con la organización del trabajo en puestos aislados en los que la interacción con otro puesto es recibir una entrada para dar una salida. El trabajo en equipo requiere habilidades y competencias propias de dicha forma de trabajo por lo que su utilización requiere que las definiciones de los puestos de trabajo estén actualizadas de acuerdo con estas nuevas necesidades. Un ejemplo de esta forma de organización del trabajo son las metodologías de desarrollo ágiles (Cockburn & Highsmith, 2001), cuya utilización va en aumento (Dyba & Dingsoyr, 2009). Otra de las diferencias es la utilización de una

estructura organizacional distribuida a lo largo del planeta, denominada *Global Software Development* (GSD) (Herbsleb & Moitra, 2001; Hernández-López, Colomo-Palacios, García-Crespo, & Soto-Acosta, 2010); esta estructura de trabajo tiene como objetivos desarrollar software de forma ininterrumpida y dar mantenimiento continuado. Desde el punto de vista del puesto, esta forma de organización requiere una serie de cambios en los puestos de trabajo: multiculturalidad (Linna, Karttunen, & Jaakkola, 2011), idiomas (Abufardeh & Magel, 2010), gestión de la confianza (Moe & Šmite, 2008)... Estos cambios hacen que los puestos de trabajo en IS, y en este caso concreto los relacionados con los proyectos de desarrollo de software, difieran entre cada organización y sea necesario disponer de una definición actualizada ellos, máxime cuando se quiere medir la productividad a nivel de puesto de trabajo.

2.4.2. Puestos de trabajo según Métrica 3

Dentro de la metodología Métrica 3, la cual ofrece a las organizaciones un instrumento útil para la sistematización de las actividades que dan soporte al ciclo de vida del software (2000), se definen los siguientes perfiles: *Directivo*, *Jefe de Proyecto*, *Consultor*, *Analista*, y *Programador*. Dentro de estos perfiles se encuentran varios participantes:

- *Directivo*: Comité de Dirección, Comité de Seguimiento, Directores de Usuarios, y Usuarios Expertos.
- *Jefe de Proyecto*: Jefe de Proyecto, Responsable de Implantación, Responsable de Mantenimiento, Responsable de Operación, Responsable de Sistemas, Responsable de Seguridad, y Responsable de Calidad.
- *Consultor*: Consultor, Consultor Informático, Consultor de las Tecnologías de la Información, Consultor de Sistemas de Información, Especialista en Comunicaciones, Técnico de Sistemas, y Técnico de Comunicaciones.
- *Analista*: Analista, Administrador de Base de Datos, Equipo de Arquitectura, Equipo de Formación, Equipo de Implantación, Equipo de

Operación, Equipo de Seguridad, Equipo de Soporte Técnico, Equipo de Proyecto, y Grupo de Aseguramiento de la Calidad.

- *Programador*: Programador.

Los participantes (o puestos de trabajo) más vinculados a los proyectos de desarrollo de software dentro de Métrica 3 son:

- **Jefe de Proyecto**, cuya función la define como: *"El Jefe de Proyecto realiza la estimación del esfuerzo necesario para llevar a cabo el proyecto, selecciona la estrategia de desarrollo, determina la estructura del mismo seleccionando los procesos principales de Métrica Versión 3 que lo integran, fija el calendario de hitos y entregas y establece la planificación del proyecto. Es el encargado de dirigir el proyecto, realizando las labores de seguimiento y control del mismo, revisión y evaluación de resultados y coordinación del equipo de proyecto. Se ocupa también de la gestión y resolución de incidencias que puedan surgir durante el desarrollo del proyecto así como de la actualización de la planificación inicial. Entre sus funciones se encuentran la elaboración de los informes de seguimiento y el archivo de la documentación de gestión del proyecto una vez que este ha finalizado."*
- **Consultor**, cuya función la define como: *"[...] asesorar en las cuestiones sobre las que tienen un conocimiento especializado. Se diferencia así entre Consultor, que asesora en los aspectos relativos al negocio y Consultor Informático, con un nivel de especialización mayor en los aspectos relacionados con la informática, su aplicación e integración en la organización. En el ámbito de la Consultoría Informática se distingue entre Tecnologías de la Información y Sistemas de Información. El Consultor en Tecnologías de la Información aporta un mayor conocimiento de las últimas tecnologías, colabora en la evaluación de distintas alternativas tecnológicas y participa en la validación y selección de la solución más*

adecuada para el sistema a desarrollar, mientras que el Consultor de Sistemas de Información ofrece una opinión experta, pericia o conocimientos relativos a los requisitos del negocio, técnicos y de usuario que han de tenerse en cuenta en el desarrollo de un sistema de información."

- **Analista**, cuya responsabilidad la define como: *"[...] elaborar un catálogo detallado de requisitos que permita describir con precisión el sistema de información, para lo cual mantendrán entrevistas y sesiones de trabajo con los responsables de la organización y usuarios, actuando de el interlocutor entre éstos y el equipo de proyecto en lo que a requerimientos se refiere. Estos requisitos permiten a los analistas elaborar los distintos modelos que sirven de base para el diseño, obteniendo los modelos de datos y de procesos en el caso del análisis estructurado y los modelos de clases e interacción de objetos en análisis orientado a objeto. Así mismo realizan la especificación de las interfaces entre el sistema y el usuario."*
- **Programador**, cuya función la define como: *"[...] construir el código que dará lugar al producto resultante en base al diseño técnico realizado por el analista o analista programador, generando también el código asociado a los procedimientos de migración y carga inicial de datos. Igualmente se encarga de la realización de las pruebas unitarias y participa en las pruebas de conjunto de la aplicación."*

Estos puestos de trabajo pueden ser asimilados a otros puestos en O*Net, de forma que se tiene una definición de estos puestos de trabajo más completa que la que aporta Métrica 3:

- **Jefe de Proyecto - Computer and Information Systems Manager** (11-3021.00) - *"Plan, direct, or coordinate activities in such fields as electronic data processing, information systems, systems analysis, and computer programming."*

- **Consultor - Computer Systems Analyst** (15-1121.00) - *"Analyze science, engineering, business, and other data processing problems to implement and improve computer systems. Analyze user requirements, procedures, and problems to automate or improve existing systems and review computer system capabilities, workflow, and scheduling limitations. May analyze or recommend commercially available software."*
- **Analista - Software developers** (15-1132.00) - *"Develop, create, and modify general computer applications software or specialized utility programs. Analyze user needs and develop software solutions. Design software or customize software for client use with the aim of optimizing operational efficiency. May analyze and design databases within an application area, working individually or coordinating database development as part of a team. May supervise computer programmers."*
- **Programador - Computer Programmer** (15-1131.00) - *"Create, modify, and test the code, forms, and script that allow computer applications to run. Work from specifications drawn up by software developers or other individuals. May assist software developers by analyzing user needs and designing software solutions. May develop and write computer programs to store, locate, and retrieve specific documents, data, and information."*

Cada uno de estos puestos tiene definidos una serie de elementos (tareas, tecnologías y herramientas, conocimiento, habilidades, capacidades, entorno de trabajo, actividades de de trabajo, formación necesaria, intereses, competencias, y valores) que ayudan a definir con mayor precisión estos puestos de trabajo de forma más precisa. Estos elementos pueden ser consultados para más información en la web de O*Net y pueden utilizarse los códigos indicados (código entre paréntesis en la lista anterior) o el nombre del puesto para acceder a dichos puestos utilizando el buscador que O*Net facilita: <http://www.onetonline.org/find/>

2.5. Medición de la productividad a nivel de puesto de trabajo en proyectos de desarrollo de software

La investigación sobre medición de la productividad en IS se ha centrado principalmente en niveles de medición superiores al del puesto de trabajo. Por ejemplo, es fácil encontrar estudios a nivel de sector e incluso nivel de país (Tsunoda, Monden, Yadohisa, Kikuchi, & Matsumoto, 2009), a nivel de la organización (Anda, Sjoberg, & Mockus, 2009) y a nivel de proyecto (Kitchenham & Mendes, 2004). Sin embargo, a niveles inferiores (equipo de trabajo y puesto de trabajo) no han tenido el mismo interés (Petersen, 2011), quizá por la dificultad de medir a dicho nivel (Ramirez & Nembhard, 2004). Hay que destacar que las medidas empleadas a nivel de puesto de trabajo son las mismas que las empleadas a niveles superiores de medición (Petersen, 2011). Esto representa un problema ya que los recursos empleados (entradas), los productos y/o servicios generados (salidas) y los factores que afectan a la productividad a este nivel no son los mismos que a niveles superiores, de modo que la eficacia de las medidas empleadas en este nivel está cuestionada (Briand, Morasca, & Basili, 2002). Además, a pesar de que ciertas medidas de productividad son imperfectas se continúan utilizando dada la facilidad de su uso (Kitchenham & Mendes, 2004), lo que añade mayor controversia a la utilización de las mismas medidas en distintos niveles de medición.

La medición de la productividad a nivel de puesto de trabajo en IS tiene como origen los trabajos publicados a finales de los años 70 y comienzos de los 80 (Jones, 1981; Thadhani, 1984; Walston & Felix, 1977). En sus comienzos, la medición de la productividad, al igual que otros tipos de medidas, se centraba en la actividad de programación (Chrysler, 1978). Por otro lado, en esos años, otros autores indicaban la importancia del factor humano en el desarrollo software (DeMarco & Lister, 1985, 1987). En aquel contexto se crearon medidas de productividad basadas en la filosofía

ingenieril que utiliza la productividad como sinónimo de eficiencia de recursos. De este modo, se utilizaron medidas basadas en SLOC y posteriormente, PF tras su definición y creación (Albrecht & Gaffney, 1983), basadas en PF. Estas medidas no reflejan toda la actividad desarrollada por cada puesto de trabajo (Petersen, 2011), sino que son medidas de productividad que tienen por objetivo medir la eficiencia de la entrega del proyecto.

Una vez han sido tratados los factores, y las entradas y salidas, es posible abordar las medidas de productividad a este nivel de medición. Teniendo en cuenta la problemática descrita en los apartados anteriores, tiene sentido plantearse si las medidas de productividad empleadas actualmente en IS eficaces para medir la productividad de cada uno de los puestos de trabajo. La respuesta será afirmativa si la medida mide lo que el puesto de trabajo debería aportar y lo que debería utilizar para realizar dicho aporte (Koch & McGrath, 1996). De este modo, las medidas clásicas (SLOC/t o PF/t) no parecen ser eficaces para todos los puestos de trabajo en IS. Puede que sean de gran utilidad en puestos de generación de código (programador) pero actualmente no todos los desarrollos emplean código nuevo (Mohagheghi & Conradi, 2007), sino que se reutiliza código y por lo tanto emplear medidas basadas en las SLOC producidas puede ser poco eficaces incluso para dichos puestos (Banker, Datar, & Kemerer, 1987). Por otro lado, y utilizando el mismo puesto de trabajo (programador), hay que tener en cuenta las tareas de mantenimiento de software en las que las líneas de código generadas no es tanto la finalidad del puesto sino la solución de errores o la inclusión de nueva funcionalidad (ISO, 2006).

Considerando las observaciones anteriores, y volviendo al ejemplo del analista definido en Métrica 3, la medida de productividad a emplear para dicho puesto sería del tipo:

$$p_1 = f(s_1(\text{catálogo de requisitos}), e_1(\text{conocimiento}), e_2(\text{usuarios}), \text{factores})$$

$p_2 = f(s_2(\text{modelos}), e_3(\text{catálogo de requisitos}), e_4(\text{usuarios}), e_5(\text{otros puestos}),$
 $\text{factores})$

$$pp = g(p_1, p_2, \text{factores})$$

donde

f es una medida que emplea las entradas y salidas para dar un resultado de productividad,

p_x es una medida de productividad para una tarea del puesto de trabajo (p_x) que emplea como medidas de entradas (e_x) y salidas (s_x) y devuelve un resultado,

g es una combinación de medidas de productividad para el puesto que devuelve un resultado de productividad influenciado por una serie de factores para el puesto de trabajo (pp),

s_x es una medida de una salida,

e_x es una medida de una entrada,

factores son todos los factores que afectan a la productividad en el puesto.

Pese a que la definición textual es sencilla, la dificultad radica en establecer unidades de medición para cada elemento medido y en definir una unidad de productividad que permita incluir las medidas de cada elemento en cada función de productividad. Por otro lado, los factores, incluida la calidad, deben ser evaluados para ver su importancia y signo en la función de productividad.

2.6. Revisión sistemática de la literatura

De forma paralela a la construcción del estado de la cuestión mediante la vía convencional de revisión de literatura manual, existe otra forma de construirlo mediante la realización de una Revisión Sistemática de la Literatura (“*Systematic Literature Review*”, *SLR*), o también denominado de forma resumida Revisión Sistemática. Una SLR es un medio de identificación, evaluación e interpretación de toda la investigación disponible relevante para una pregunta de investigación concreta, un área temática, o fenómeno de interés. Este tipo de estudios ha sido ampliamente utilizado en otras áreas de conocimiento, por ejemplo medicina y las ciencias sociales ante la necesidad de disponer de visiones conjuntas sobre elementos concretos (Fink, 2005; Petticrew & Roberts, 2005). Aunque en la actualidad existen diferentes referencias que explican el proceso de diseño y ejecución de una SLR, para la realización de la presente SLR se ha optado por emplear como elemento de referencia el manual desarrollado por Kitchenham (2007). Esta elección se debe a la vinculación de la autora con el área de IS, y por ser una persona referente en la utilización de este método dentro del área de investigación.

Un primer paso para la realización de una SLR es analizar la literatura para localizar SLR previas con objetivos de conocimiento similares. Si se consiguen resultados positivos, se dispone entonces de estrategias de búsqueda para objetivos similares. Además, en caso de tener los mismos objetivos de investigación, se dispondría de una SLR previa que podría ser replicada ampliándola con las publicaciones relacionadas hasta la fecha. Este paso no ha dado resultados positivos por lo que no hay una estrategia de búsqueda previamente publicada para realizar una SLR entorno a la medición de la productividad en IS⁶. No obstante, han sido

⁶ Posteriormente a la ejecución de la SLR se ha encontrado una revisión con objetivos más globales pero dentro del área de conocimiento (productividad dentro del área de desarrollo de software), publicada

encontradas dos SLR existentes relativas a la productividad en la IS, pero su objetivo es revisar los factores que afectan a la productividad (Paiva, et al., 2010; Wagner & Ruhe, 2008b). De forma más general tampoco se ha encontrado una SLR entorno a la medición de la productividad en trabajos con fuerte carga de conocimiento y humana, aunque existen otros proyectos exploratorios sin metodología, como por ejemplo una taxonomía de medición de productividad (Ramirez & Nembhard, 2004). Así pues, la estrategia de búsqueda ha tenido que ser diseñada desde cero para la presente investigación.

Para llevar a cabo una SLR es necesario seguir una serie de actividades que pueden ser agrupadas en tres fases: planificación, ejecución y comunicación. El proceso de trabajo a llevar en una SLR se detalla en un protocolo; en este caso puede consultarse el protocolo de la SLR realizada en el Anexo A - Protocolo revisión sistemática. El protocolo, una vez creado, pasó un proceso de revisión llevado a cabo por D. David Aguado (tutor de trabajo fin de Máster en la UAM) y D. Ricardo Colomo de forma paralela e independiente. Una vez obtenido el *feedback* de cada uno de ellos, se realizó una versión final que fue revisada por D. David Aguado. No obstante, a continuación se presenta el proceso diseñado y que finalmente se ha ejecutado de forma resumida, y teniendo en cuenta las modificaciones realizadas, ya que el protocolo inicialmente diseñado evolucionó sobre la marcha debido a que los resultados no se ajustaban a las necesidades investigadoras. En primer lugar se explican las fuentes de información utilizadas sobre las que se ejecutará la cadena de búsqueda definida; en segundo lugar, se explica la cadena de búsqueda; en tercer lugar, se explica el proceso de búsqueda junto con los criterios de inclusión y exclusión para seleccionar artículos de interés en este proyecto; y en cuarto lugar, se explica el proceso de extracción de datos centrado en los objetivos del presente proyecto.

en las mismas fechas en las que se ejecutó la búsqueda: Petersen, K. (2011). Measuring and predicting software productivity: A systematic map and review. *Information and Software Technology*, 53(4), 317-343.

2.6.1. Preguntas de investigación

Basándose en la literatura analizada previamente a la realización del protocolo, y teniendo en cuenta las hipótesis de investigación planteadas, se establecen las siguientes preguntas de investigación como objetivo de esta SLR:

P. 1. ¿Cuáles son las entradas y salidas del proceso productivo interno de la ingeniería del software?

P. 2. ¿Son distintas las entradas y salidas para los diversos puestos de trabajo dentro de la ingeniería del software?

Si la respuesta es afirmativa, entonces se plantean las siguientes preguntas:

P.2.1. ¿Cuáles son las entradas y salidas para el ingeniero de software?

P.2.2. ¿Cuáles son las entradas y salidas para el programador?

P.2.3. ¿Cuáles son las entradas y salidas para el analista?

P.2.4. ¿Cuáles son las entradas y salidas para el jefe de proyecto?

2.6.2. Fuentes de consulta

Dada la diversidad de fuentes para consultar artículos en formato electrónico vía web mediante los acuerdos firmados por la UAM y de la UC3M con diversas revistas y editoriales, se establecen las siguientes fuentes de consulta:

- IEEEExplore.
- ScienceDirect.
- ISI Web of Science (WoS).
- Wiley Online.
- ACM Digital Library.

- Taylor & Francis.
- SpringerLink⁷.

2.6.3. Cadena de búsqueda y delimitaciones temporales

Una vez seleccionadas las fuentes de información, se define la cadena de búsqueda partiendo de la población de estudio, y de palabras clave y sinónimos. Por un lado, la población objeto de estudio son las entradas (p. ej. horas-hombre, conocimiento...) y salidas (p. ej. SLOC, PF, requisitos, diseños, documentación...) de los puestos de trabajo relativos al proceso productivo de la IS (p. ej. ingeniero software, programador, jefe de proyecto, analista, diseñador...). Con esta población la lista de palabras clave y sus sinónimos, junto con su traducción al inglés (idioma en el que se ejecutará la búsqueda), utilizada para generar la cadena de búsqueda, fue la siguiente:

- Productividad: desempeño.
 - Productivity: performance.
- Entrada: recurso.
 - Input: resource.
- Salida: producto, servicio.
 - Output: product, service.
- Personal: empleados.
 - Personnel: staff.
- IS: desarrollo de software, mantenimiento de software.
 - Software engineering: software development, software maintenance.

⁷ Tuvo que ser descartado dado que no es posible ejecutar una cadena de búsqueda de gran tamaño, debido a una limitación de caracteres.

El año de inicio será 1981 dado que a partir de ese año se empiezan a publicar las investigaciones más referenciados sobre mediciones de esfuerzos y tamaños en IS (Albrecht & Gaffney, 1983; Boehm, 1981b; DeMarco, 1986). Estas investigaciones crearon un marco de trabajo en los años posteriores para las investigaciones relacionadas, entre ellas, la productividad.

Hay que destacar que los sinónimos de IS seleccionados son dos grandes áreas dentro de la ingeniería que representan gran parte de la actividad y son aquellas en las que los puestos de trabajo bajo estudio tienen influencia directa. Para generar la cadena de búsqueda se utiliza un lenguaje booleano con operadores AND y OR, y comillas para texto exacto. Este formato de cadena está reconocido por todas las fuentes de información utilizadas. Así pues, la cadena de búsqueda finalmente utilizada es la siguiente:

("software engineering" OR "software development" OR "software maintenance") AND (productivity OR performance) AND (input OR resource OR output OR product OR service) AND (personnel OR staff)

2.6.4. Proceso de recuperación, filtrado y revisión

El procedimiento para recuperar, filtrar y revisar estudios es el siguiente: en primer lugar, se ejecuta la cadena de búsqueda en los distintos motores de búsqueda; en segundo lugar, para seleccionar un conjunto inicial de estudios, los resúmenes de todos los estudios recuperados son leídos y evaluados de acuerdo con los criterios de inclusión y exclusión (ver Anexo A - Protocolo revisión sistemática); en tercer lugar, para afinar el conjunto inicial de estudios, se recupera y lee cada artículo entero para verificar su inclusión o exclusión. El motivo de exclusión o inclusión en esta tercera etapa se documenta, mientras que para la etapa anterior no se realiza esta documentación; en cuarto lugar, una vez seleccionados estos estudios primarios, se analizan sus referencias para identificar posibles estudios que no hayan sido encontrados en la búsqueda y que deban ser analizados en esta revisión sistemática.

Este procedimiento (ver Ilustración 8) está basado en el utilizado en una revisión sistemática reciente (Afzal, Torkar, & Feldt, 2009).

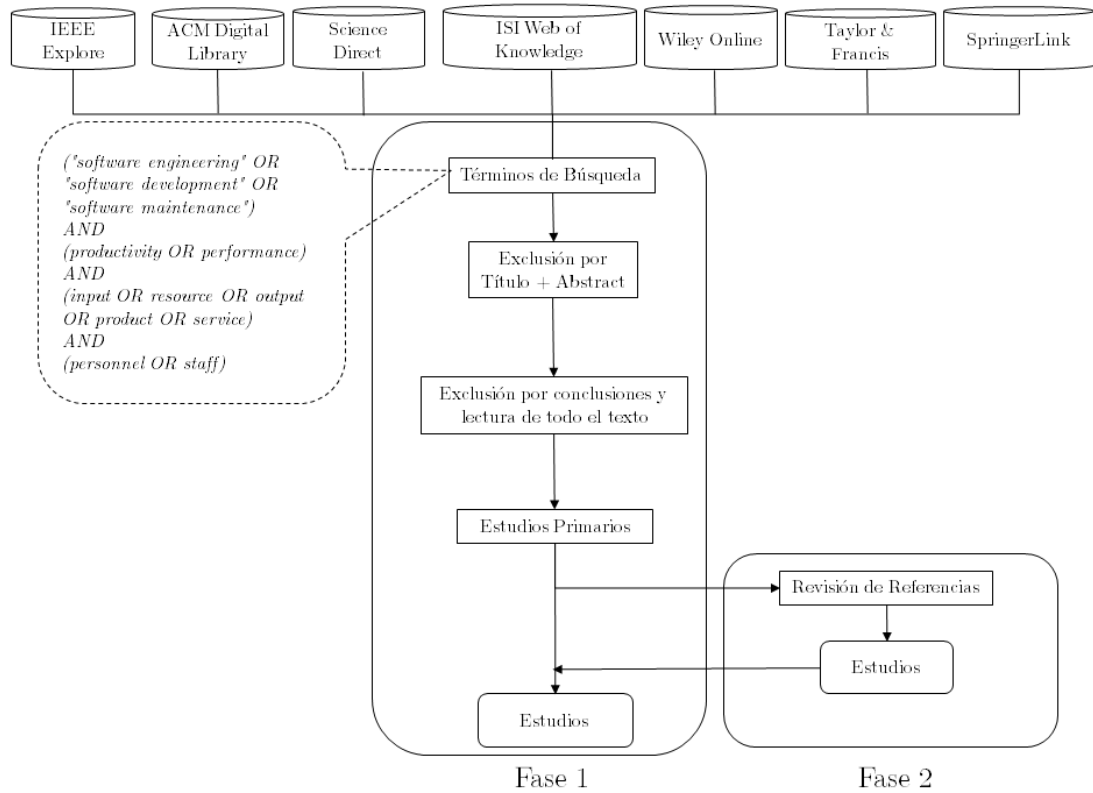


Ilustración 8. Proceso de filtrado; basado en (Afzal, et al., 2009)

Todo el proceso de búsqueda se documenta y los elementos se almacenan en un gestor de referencias. En dicho gestor, se crean grupos de elementos en función del resultado que se obtienen en cada filtro. Para los elementos que superan el primer filtro fueron se recupera el texto completo, que también se almacena en el gestor de referencias. Para los elementos finales, se rellena una ficha con los datos necesarios para su posterior análisis, tal y como se explica a en el Anexo A - Protocolo revisión sistemática.

2.6.5. Ejecución

Los estudios incluidos y excluidos se presentan por fases siguiendo el proceso de búsqueda descrito anteriormente. Dada la longitud de algunas de las listas de

referencias, estas se han añadido en el Anexo B - Listas de resultados de la revisión sistemática.

2.6.5.1. Estudios incluidos y excluidos - Fase 1

La primera fase del proceso de búsqueda fue ejecutada el 16/05/2011, y los resultados fueron los siguientes:

Fuente	N
IEEEExplore	110
ACM Digital Library	5 ⁸
ScienceDirect	11
SrpingerLink	0
Wiley Online	38 ⁹
ISI WoK	13
Taylor & Francis	0
Total	187
Total (Sin repetidas)	177

Tabla 5. Resultados Fase 1 – Sin filtrar (16/05/2011)

De los 187 resultados obtenidos, 10 eran elementos duplicados; en concreto, estos elementos duplicados fueron resultados de ISI WoK. De estos 177 resultados, 22 fueron descartados por ser resultados incompletos o no relacionados con la IS. De los 155 restantes, 98 fueron excluidos tras la lectura del título y el resumen; así pues, 120 resultados fueron excluidos en el primer filtro, por lo que quedaron 51 resultados para

⁸ Otros 5 resultados eran enlaces a *issues* directamente, por lo que no han sido incluidos como resultados de la búsqueda.

⁹ La cadena de búsqueda ha sido ejecutada de la siguiente manera ya que el buscador no permitía ejecutarla de forma análoga al resto de buscadores: TITLE =("software engineering" OR "software development" OR "software maintenance") AND (productivity OR performance) AND (input OR resource OR output OR product OR service) AND (personnel OR staff) OR ABSTRACT=("software engineering" OR "software development" OR "software maintenance") AND (productivity OR performance) AND (input OR resource OR output OR product OR service) AND (personnel OR staff). Por lo que puede que los resultados se hayan visto afectados.

filtrar con la lectura de texto completo utilizando los criterios de inclusión y exclusión. De los 51 resultados útiles para el siguiente filtro, 27 son artículos de revistas y 24 de conferencias. La lista con los resultados sin repetir y con los subgrupos de incluidos y excluidos en este filtro puede ser consultada en el Anexo B - Listas de resultados de la revisión sistemática, en el apartado Fase 1. Lista Inicial.

Segundo filtro (inclusión/exclusión por lectura de texto completo)

En este segundo filtro, si el resultado no cumplió los criterios de exclusión fue incluido. Tras la lectura del texto completo de cada uno de ellos, los resultados fueron los siguientes:

Excluidos	48
Incluidos	3
Total	51

Tabla 6. Resultados Fase 1 – Primer Filtro (16/05/2011)

La lista con el resultado del filtro para cada uno de los elementos puede ser consultada en el Anexo B - Listas de resultados de la revisión sistemática, en el apartado Fase 1. Resultados tras lectura de texto completo. Una vez terminada la primera fase, los resultados obtenidos son tres; entre estos resultados hay un artículo de revista y dos artículos de conferencias. Tras esta primera fase, se ejecutó la Fase 2 que consiste en revisar las listas de referencias de estos resultados para obtener publicaciones relacionadas.

2.6.5.2. Estudios incluidos y excluidos - Fase 2

De los resultados finales de la Fase 1 se obtuvieron las listas de referencias y se aplicaron sobre ellas los mismos filtros aplicados anteriormente. Se obtuvieron un total de 9 referencias iniciales por la lectura del título y *abstract*. De estas referencias 3 fueron finalmente seleccionadas aplicando los criterios de inclusión y exclusión. Esta nueva lista con resultados para cada uno de los elementos puede ser consultada en el

Anexo B - Listas de resultados de la revisión sistemática, en el apartado Fase 2. Resultados tras lectura de texto completo.

Así pues, con estos resultados se han obtenido un total de 6 referencias tal y como se muestra en la siguiente tabla:

Resultado Fase 1	3
Resultado Fase 2	3
Total	6

Tabla 7. Resultados tras ejecutar la Fase 2

2.6.6. Resultados

En total se han obtenido seis resultados tras la ejecución de ambas fases de la revisión sistemática; de estos resultados, cuatro son artículos de conferencias y dos de revistas. De forma resumida los resultados se resumen en la Tabla 8.

Categoría	Ítems	N
Entradas utilizadas	Unidad de tiempo	5
	Tareas asignadas	1
Salidas utilizadas	SLOC	3
	Tareas completadas	3
	Valoración (dificultad y esfuerzo)	1
Formula de productividad	SLOC/tiempo	3
	Tareas/tiempo	3
Tipo de fuente de datos	Universitaria	3
	Organizacional	2
	Simulación	1

Tabla 8. Resultados cuantitativos

Por otro lado, llegado este punto es posible dar respuesta a las preguntas de investigación planteadas:

P.1. ¿Cuáles son las entradas y salidas del proceso productivo interno de la ingeniería del software?

Por un lado, las entradas utilizadas son unidades de tiempo, principalmente horas y jornadas de trabajo, y tareas asignadas al trabajador. Por otro lado, las salidas son líneas de código y tareas completadas.

P.2. ¿Son distintas las entradas y salidas para los diversos puestos de trabajo dentro de la ingeniería del software?

No es posible contestar a esta pregunta con certeza ya que en todos los resultados primarios obtenidos se mide la productividad sin definir el puesto de trabajo para el cual han sido planteadas dichas medidas. Por ello, las preguntas que estaban supeditadas (cuáles son las entradas y salidas para cada puesto de trabajo), a una respuesta afirmativa a esta pregunta, no tienen respuesta.

2.6.7. Hallazgos

A continuación se presentan los hallazgos destacados a partir de los resultados de la SLR. De forma resumida se puede decir que los resultados indican una dualidad en las salidas a medir y la fórmula de productividad a emplear. Sin embargo, las entradas utilizadas son las mismas en casi todos los resultados: se emplea la unidad de tiempo en cinco de los seis resultados; y las fuentes de datos empleadas no son del todo similares.

Con respecto a las entradas, existe una clara tendencia a emplear unidades de tiempo como medida (horas, días, semanas...), aunque uno de los resultados emplea las tareas asignadas como medida de entrada (Kiebert et al., 1996). El resultado que utiliza las tareas asignadas utiliza tareas iguales para todos los participantes en el estudio, es decir, todos los participantes realizan las mismas tareas. Esta peculiaridad es prácticamente imposible de extrapolar al entorno real de trabajo de la IS, debido a que cada tarea es única y se realiza, salvo re-trabajos, una única vez. No obstante, hay

que tener en cuenta que esta entrada es de distinta tipología a la unidad de tiempo; una tarea es más compleja de medir y seguramente tenga, entre otras características, un tiempo de realización, un esfuerzo, una valoración de la dificultad y un riesgo asociados a ella. Así pues, esta entrada constituye un posible enriquecimiento en la medición de entradas utilizadas pese a que el estudio no puede ser llevado a un entorno real.

Los resultados referentes a la otra parte de la medición de la productividad, las salidas, están divididos en dos medidas con el mismo número de resultados (tres): SLOC y tareas completadas, que se complementan con la valoración de la tarea en estudio de Kieburtz et al. (1996); en la valoración se incluye la dificultad y el esfuerzo empleado. Este hallazgo es interesante dado que lleva a la utilización de medidas de productividad que no utilizan, al menos en su totalidad, medidas de tamaño (líneas de código, funcionalidad...) como salida sino que la realización de las tareas en sí mismas es la salida que cada trabajador realiza. Por otro lado, la utilización de una medida de tamaño como son las SLOC no resulta sorprendente ya que es una de las unidades de salida ampliamente utilizadas en distintos niveles de medición de productividad e incluso para medidas relacionadas con la estimación. La popularidad de esta medida radica en su fácil medición, ya que es casi automática, aunque establecer una medida común basada en SLOC es complicado. Sin embargo, según DeMarco (1986), el problema de utilizar las líneas de código como salida no es tanto la carencia de una medida común sino su tardía aplicabilidad. Además, diversas tareas dentro de la realización de proyectos de desarrollo de software, por ejemplo la especificación de requisitos o el diseño, son difícilmente convertibles en SLOC u otra medida de tamaño.

Una vez presentados los resultados para las entradas y salidas, los resultados para la fórmula empleada para medir la productividad sólo pueden ser combinaciones entre ellas. De este modo, los resultados se dividen en dos grupos idénticos que los de las salidas: SLOC/Tiempo y Tareas/Tiempo. Hay que destacar, que aunque un resultado

definía como entrada las tareas asignadas, a la hora de medir la productividad empleaba la unidad de tiempo como medida. Por lo que podría decirse que todas las entradas detectadas son medidas de tiempo en realidad. Es importante destacar que todas las medidas de productividad encontradas se basan en la visión de la eficiencia del trabajador durante el tiempo contratado lo que limita la visión de la medición de la productividad a la eficiencia (Koss & Lewis, 1993). Además, estas medidas no permiten, al menos en la forma en la que se plantean, personalizar las medidas a las necesidades organizacionales y culturales de cada organización y entorno de trabajo. Además, estas medidas son medidas de productividad parciales ya que no tienen en cuenta todas las entradas ni todas las salidas (Craig & Harris, 1973; Dale & van der Zee, 1992). De este modo, ninguno de los resultados primarios cubre la necesidad de disponer de medidas totales de productividad por puesto de trabajo que sean adaptables a cada cultura, organización, y puesto de trabajo.

Hay que destacar que no existe una diferenciación clara por puesto de trabajo en ninguno de los estudios obtenidos. Considerando las salidas empleadas en dichos estudios (SLOC y tareas completadas) la aplicabilidad de las medidas se centra en dos tipos de puestos de trabajo. Por un lado, las medidas basadas en SLOC, tanto para las salidas como para la productividad, pueden ser aplicadas exclusivamente a puestos en los que el trabajador produzca de forma directa dicha salida, principalmente este puesto es el de programador y de forma más general ingeniero de software. Por otro lado, las medidas basadas en tareas completadas son más generales y pueden ser aplicadas a infinidad de puestos en los que exista una estructuración del trabajo en tareas claramente identificables y medibles.

Pese a que en la segunda fase se descartaron tres resultados durante su lectura de texto completo, es importante resaltar algunas observaciones sobre ellos. Estos resultados son libros de referencia en el área de IS y por lo tanto son muy utilizados como fuente de conocimiento y formación (Brooks Jr., 1985; DeMarco, 1986; Sommerville, 2010). Estos libros incluyen referencias a la productividad tanto para su medición como en lo referente a los factores que la afectan. Por ejemplo, Sommerville

(2010) da la siguiente definición de productividad: “*Software productivity is an estimate of the average amount of development work that software engineers complete in a week or a month. It is expressed as lines of code/month, function points/month, etc.*”. Ésta definición está en línea con los resultados de la revisión sistemática realizada ya que utiliza una medida del producto como salida, una medida de tiempo como entrada y un ratio como fórmula de cálculo. Además, Sommerville (2010) comenta la dificultad para medir la productividad ya que las salidas y entradas no son del todo tangibles, a lo que añade la influencia de la calidad en el proceso productivo con la dificultad inherente de medir su influencia en la productividad. Por otro lado, DeMarco (1986) indica que las diferencias en productividad entre los trabajadores pueden ser enormes, tal y como apuntan los estudios previos realizados por otros investigadores (Boehm, 1981a; Myers, 1975; Sackman, Erikson, & Grant, 1968), diferencia que puede verse incrementada si el tamaño del equipo de trabajo aumenta. Además, DeMarco (1986) plantea que el tiempo y las personas, como entradas al proceso productivo, no son intercambiables en los modelos de productividad aunque existe cierto grado de intercambio entre estas dos entradas. Y añade una regla de mejora de productividad en el desarrollo de software con base en los resultados de estudios previos: “Excluir del equipo a alguien con muy bajo rendimiento puede ser a menudo más productivo que añadir a alguien con buen rendimiento”¹⁰.

Finalmente hay que destacar que la tipología de las fuentes de datos empleadas en cada uno de los estudios primarios está en línea con la tendencia en la experimentación con participantes universitarios pese a que el objeto de investigación sean los trabajadores del sector (Kitchenham et al., 2002). De forma adicional, destacar que un resultado emplea una fuente de datos de tipo simulador ya que se trata de un artículo sin participantes reales y roza lo teórico.

¹⁰ Traducción del texto: “Taking a poor performer off your team can often be more productive than adding a good one.”

2.6.8. Breve discusión de la revisión sistemática

Los resultados de la revisión sistemática indican que existe cierta unanimidad con respecto a las entradas utilizadas en la medición de la productividad. Ésta unanimidad es un reflejo de las entradas utilizadas a niveles superiores de medición, por ejemplo a nivel de organización, en los que la entrada utilizada es (casi) siempre una unidad de tiempo o esfuerzo (ver p.ej., Anselmo & Ledgard, 2003; Kitchenham & Mendes, 2004). El empleo de esta entrada en solitario a niveles inferiores, en los que el grado de detalle puede ser mayor, encajaría perfectamente en mediciones de productividad de trabajos mecanizados en los que cada tarea y operación tienen un tiempo definido y por lo tanto se planifica y mide basándose en dichos tiempos (productividad de trabajadores de cuello azul – *blue-collar productivity*). Sin embargo, en la IS los trabajadores son del tipo tradicionalmente denominado de cuello blanco (*white-collar*), los cuales emplean otros recursos que no son exclusivamente la puesta en disposición de las horas marcadas en el contrato de trabajo (Ramirez & Nembhard, 2004). Un ejemplo de entrada que utilizan estos trabajadores es el conocimiento, así como otros recursos intangibles, de ahí el nombre que Drucker (1959) creó: trabajadores del conocimiento (*knowledge workers*). No obstante, el tiempo es un recursos que se utiliza una única vez y si no se utiliza se consume igualmente, de ahí la gran importancia del tiempo para este tipo de trabajadores. Además, cuando se trabaja con unidades de tiempo puede ser necesario distinguir entre horas contratadas al trabajador y las horas que realmente se trabajan en producir las salidas. Dada la naturaleza de la IS y de otros trabajos de fuerte carga intelectual, muchos de los trabajadores realizan horas fuera de su horario para terminar las tareas asignadas ya que en cualquier momento pueden tener una idea que solucione un problema (Koss & Lewis, 1993). Así pues, las entradas utilizadas en los resultados primarios no tienen en cuenta las características propias de los trabajadores de la IS pero si tienen en cuenta la importancia del tiempo.

Las salidas por el contrario están divididas en dos grupos totalmente diferenciados. Por un lado, la utilización de las SLOC da lugar a medidas de productividad

centradas en la liberación de código fuente; las cuales pueden ser de gran interés en determinados entornos como por ejemplo en factorías de software en las que el *core business* sea la programación a partir de los diseños y requisitos facilitados. El problema de la utilización de esta salida es que no representa todas las tareas realizadas en IS y por lo tanto no es válida para todos los puestos de trabajo de dicha actividad. Por otro lado, la utilización del cumplimiento o finalización de tareas como medida de salida abre una vía paralela de medición de la productividad ya que es universal en cuanto a su uso – cualquier puesto de trabajo, generalmente en su descripción de puesto de trabajo, tiene definidas las tareas que desempeña. No obstante, el uso de esta salida conlleva la dificultad de valorar las tareas por unas determinadas medidas de valoración que pueden ser muy dispares en función de las mismas, por ejemplo la dificultad, el grado de habilidades y competencias necesarias, los recursos disponibles y utilizados... Así pues, estas salidas no dan una solución universal y por lo tanto es necesario seguir explorando alternativas que, si bien probablemente no lleguen a mejorar sustancialmente las medidas disponibles, al menos complementen las medidas actuales.

De este modo los resultados obtenidos reflejan que dentro de la IS, a nivel de puesto de trabajo, existen pocas investigaciones que traten la productividad. Además, estas investigaciones utilizan las mismas medidas utilizadas a niveles superiores de medición: SLOC/Tiempo. Estas medidas, incluso a nivel superior, no llegan a medir la productividad de una forma precisa y dejan sin medir gran parte de ella por lo que a niveles inferiores, en los que la granularidad es mayor, pierden aun más precisión e incluso su sentido (Fenton & Neil, 1999). De forma adicional, la utilización de unidades de tiempo como única entrada usada en la medición de la productividad a nivel de puesto de trabajo tiene dos lecturas: (1) el tiempo es el único recurso que es realmente gastado en la realización de tareas de conocimiento, y (2) es la única entrada tangible, precisa y universal disponible para ser empleada como medida de entrada de productividad, mientras que el resto de entradas son intangibles, imprecisas y/o no universales. Con respecto a la primera lectura, y suponiendo que el

tiempo es el único recurso gastado sería necesario disponer de tiempos estándar de producción para cada tarea y operación lo que es realmente difícil para casi todas las tareas de la IS. Esta dificultad no sucede cuando se mide la productividad de un trabajador dentro de la industria de fabricación ya que cada operación tiene un tiempo determinado para llevarse a cabo (Drucker, 1999). Con respecto a la segunda, la disponibilidad de esta medida no tiene porque ser exclusiva, es decir, puede combinarse con otras unidades de medida para generar medidas de productividad alternativas. Así pues, desde el punto de vista del autor de la investigación, es necesario seguir investigando nuevas medidas de productividad para la IS a nivel de puesto de trabajo.

Los resultados obtenidos son escasos en cuanto al volumen de estudios encontrados, lo que representa una debilidad de cara a las conclusiones y observaciones que puedan obtenerse del análisis de los mismos. Esta escasez de resultados puede ser vista desde los siguientes puntos de vista:

- *Fallo en el protocolo.* Es posible que los términos de búsqueda empleados no sean lo suficientemente correctos para obtener un número mayor de resultados finales tras la ejecución de la revisión. Sin embargo, desde el punto de vista del autor de la tesis, el siguiente punto de vista parece ser más real teniendo en cuenta la revisión de la literatura convencional realizada previamente.
- *Poca literatura referente a este objetivo de investigación.* La escasez de resultados puede deberse a este punto de vista ya que generalmente la productividad se mide a niveles superiores: equipo, proyecto, organización... y no es frecuente medir la productividad a nivel de trabajador, al menos no dentro de IS.

2.6.9. Conclusiones

Como primera conclusión tras realizar esta SLR, cabe destacar que se ha generado un protocolo para la búsqueda de información sobre medición de la productividad a

nivel de puesto de trabajo dentro de la IS que puede ser utilizado en futuras revisiones sistemáticas. Además, puede ser mejorado y modificado para adaptarse a necesidades específicas dentro de esta área de investigación. Por otro lado, y con respecto a la ejecución, los estudios primarios obtenidos son escasos, pero su análisis aporta información útil para tener una visión global del estado de la cuestión que complementa la visión obtenida con la revisión clásica realizada. No obstante, sería interesante realizar una revisión sistemática que amplíe la población objetivo no limitando el área a la IS; por ejemplo una revisión sistemática a nivel de trabajadores del conocimiento o del sector servicios.

De la escasez de resultados en la SLR es posible obtener dos conclusiones específicas. Por un lado, habilita la posibilidad de realizar estudios en el área ya que no existe una saturación de la misma. Además, la baja diversidad de los elementos utilizados en los resultados apoya esta conclusión ya que no se plantean ni prueban alternativas. Quizá el principal obstáculo para el aumento de la actividad investigadora en esta área sea la dificultad para medir la productividad a este nivel, pero esto no ha de ser visto como un obstáculo sino como un reto. Por otro lado, el área podría considerarse de interés, máxime en la situación actual en la que la mayoría de los países están en una situación de turbulencias económicas y sociales. En esta situación, e incluso en épocas de bonanza económica y social, la medición de la productividad (con el objetivo de su mejora) generalmente está bien vista. Aunque, como comenta DeMarco (1986), medirla de forma individualizada no está bien visto por la mayoría de las personas, y esto crea una apariencia de fascismo corporativo, lo que provoca que no se mida de forma individualizada por miedo a que los trabajadores se sientan insultados y se marchen. Aunque según sus conclusiones, es necesario disponer de esta información para asignar correctamente tareas a los trabajadores en función de sus características, desempeño y productividad.

En segundo lugar, los resultados obtenidos no arrojan nuevas líneas de investigación ya que utilizan las mismas entradas y salidas, y por lo tanto las mismas

medidas de productividad que se emplean a niveles superiores de medición (proyecto, departamento, organización...). Como entradas, la utilización de medidas de tiempo es la principal elegida en las investigaciones de productividad a nivel de puesto de trabajo. Como salidas, las SLOC y la realización de tarea. De este modo, las medidas de productividad a nivel de puesto de trabajo en IS tienen dos variantes: por un lado las centradas en medir la productividad de una unidad del producto (SLOC/t) y por otro lado las centradas en unidades de planificación del proyecto (Tareas Completadas/t). Las medidas centradas en la productividad de unidades del producto, que principalmente tienen el objetivo de medir la eficiencia de poner en explotación un producto, son las que se emplean principalmente a niveles superiores en IS. De forma alternativa a las líneas de código en dichos niveles se emplean medidas tales como Puntos Función que miden la funcionalidad que el producto software ofrece. Por otro lado, las medidas centradas en unidades de planificación del proyecto se emplean con menor frecuencia en niveles superiores de medición de la productividad.

En tercer lugar, es alarmante la ausencia de factores relacionados con la calidad en la medición de la productividad. Con respecto a este elemento, Drucker (1999) indica que la calidad es un factor clave para determinar la productividad de un trabajador del conocimiento, ya que la productividad de estos trabajadores es una cuestión de calidad, y no sólo de cantidad. Además, es necesaria la inclusión de otras dimensiones principalmente la satisfacción del cliente, la responsabilidad e importancia del trabajo, la percepción de la productividad por parte de los trabajadores y el absentismo (Ramirez & Nembhard, 2004). Paralelamente a estas otras dimensiones, algunas específicas a la IS podrían ser consideradas. Por ejemplo la reutilización del código fuente y otros productos del proceso productivo, el efecto del entorno de trabajo, las capacidades y experiencias del trabajador, la cultura organizacional o las características del proyecto. La inclusión de estas dimensiones aportaría nuevas medidas de productividad más acordes con la casuística de los puestos de trabajo bajo estudio.

Finalmente, señalar que los resultados de esta SLR han sido publicados recientemente, tal y como se puede comprobar en el anexo correspondiente a las contribuciones realizadas a la literatura.

2.7. Resumen del estado de la cuestión

La productividad en IS ha sido ampliamente estudiada a nivel de proyecto y organización (Petersen, 2011), pero existe escasa investigación a nivel de puesto de trabajo más utilizadas (Hernández-López, et al., 2013). En concreto, las medidas de productividad empleadas se centran en la relación entre una salida y una entrada, utilizando ratios y/o regresiones lineales, en línea con la definición aportada por la norma IEEE 1045-1992. Esta norma define la productividad como la relación de una primitiva de salida (p. ej., líneas de código, puntos función o documentos) y su correspondiente primitiva de entrada (p. ej., esfuerzo, tiempo) para desarrollar software. En algunos casos se utilizan varias entradas pero la metodología empleada suele ser la misma: ratios o regresiones fijando una sola salida (ver p. ej.: Kitchenham & Mendes, 2004).

En la SLR realizada se ha obtenido una visión sobre las entradas y salidas utilizadas en las medición de la productividad a nivel de puesto de trabajo (Hernández-López, et al., 2013). Los resultados de esta revisión indican que hay cierta unanimidad con respecto a las entradas utilizadas en la medición de la productividad a dicho nivel. Esta circunstancia puede deberse a que las entradas normalmente utilizadas en niveles superiores de medición se centran en el tiempo y/o el esfuerzo (e.g., Anselmo & Ledgard, 2003; Kitchenham & Mendes, 2004). El uso de estas entradas como las únicas entradas a niveles inferiores encaja perfectamente en los sectores manufactureros, pero los proyectos de desarrollo de software engloban actividades con fuerte carga de capital humano en detrimento de la mano de obra centrada en la fuerza física. En los sectores manufactureros, cada tarea tiene un

tiempo definido para cada operación y de este modo la planificación y medición está basada en esas medidas. Sin embargo en IS no se dispone de esa facilidad.

Los trabajadores de IS pueden ser considerados trabajadores del conocimiento o trabajadores de cuello blanco (Drucker, 1999; Ramirez & Nemhard, 2004). Ejemplos de entradas utilizadas por estos trabajadores son el conocimiento, la experiencia, la colaboración con otros trabajadores, la motivación... En cualquier caso, el tiempo es un recurso que se consume igualmente tanto si se usa como sino. Además, dada la naturaleza de los trabajadores de IS, los trabajadores desempeñan horas extras fuera del trabajo y dada la tipología de las tareas desempeñadas suelen tener ideas para solucionar problemas en cualquier momento, no siempre dentro del horario de trabajo (Koss & Lewis, 1993). Así pues, el tiempo y el esfuerzo, por si mismos son unas entradas que no refleja las características de los puestos de trabajo dentro de los proyectos de desarrollo de software pero son necesarias para medir la productividad de estos profesionales.

Con respecto a las salidas, los resultados de la SLR indican que se usan dos grupos totalmente diferentes (Hernández-López, et al., 2013). Un grupo está centrado en el uso de SLOC como medida de la salida. Esta aproximación puede ser de gran interés en ciertos entornos como por ejemplo en las factorías de software en los cuales el *core business* es la codificación de los diseños y requisitos facilitados por el cliente. El otro grupo está centrado en la terminación de tareas. Este grupo plantea una manera paralela de medir la productividad y encaja en el punto de vista ingenieril de la productividad (Ghobadian & Husband, 1990). Las medidas dentro de este grupo son de carácter general dado que su utilización es universal - cualquier trabajo, habitualmente en su descripción del puesto, se compone de una serie de tareas que han de ser desempeñadas. Sin embargo, este grupo genera varias dificultades, entre ellas la dificultad de la valoración de las tareas por una medida particular que pueda variar en función de las características de cada puesto de trabajo y cada tarea.

Respecto a los puestos de trabajo en IS se puede decir que están más o menos reconocidos universalmente, aunque su definición varía en cada organización, se encuentran en constante actualización (Garcia-Crespo, et al., 2008) y su definición es algo difusa (Litecky, et al., 2010; Yilmaz, et al., 2012). En estos puestos, la utilización de capital humano, intelectual y relacional para desempeñar las tareas y cumplir los objetivos es imprescindible (Colomo-Palacios, Tovar-Caro, et al., 2010; Litecky, et al., 2010). Estas características los diferencian considerablemente de los puestos de trabajos de otras industrias en los que la presencia de la mano de obra desde un punto de vista físico, en lugar de intelectual, es el principal recurso necesario (Rus & Lindvall, 2002). Así pues, las medidas de productividad a nivel de puesto deberían tener en cuenta estas características y estar ajustadas a la dimensiones vinculadas con la productividad en ellos (Ramirez & Nembhard, 2004).

Es posible afirmar que la medición de la productividad de la productividad en IS continua siendo un reto, tal y como Drucker indicó de forma genérica para los trabajadores del conocimiento (1999). Medir la productividad de estos trabajadores representa una tarea compleja, dado que su casuística dista mucho de los puestos de trabajo que requieren casi en exclusiva capacidades físicas. En la actualidad existen varias propuestas para medir la productividad, pero ninguna cubre todas las dimensiones que definen la productividad de estos trabajadores (Ramirez & Nembhard, 2004). No obstante, existen ciertos métodos como las variantes de *Data Envelopment Analysis* (DEA) que cubren varias dimensiones de la productividad, y que pueden ser aplicados a entornos en los que se utilizan múltiples entradas y se producen múltiples salidas (Asmild, et al., 2006; Cook & Seiford, 2009; Liping, et al., 2005; Petersen, 2011; Ruan, et al., 2007; Yang & Paradi, 2004; Yang & Paradi, 2009).

Así pues, teniendo en cuenta el estado de la cuestión, el estudio de nuevas medidas de productividad para los puestos de trabajo relacionados con los proyectos de desarrollo software está justificado (Petersen, 2011). Además, para llevar a cabo este estudio es necesario conocer la definición de productividad en dicho nivel de medición,

conocer si existen otras entradas entre los recursos utilizados y otras salidas producidas, que puedan ser medidas y, por lo tanto, incluidas en las nuevas medidas de productividad.

3 Enfoque del problema

3.1. Introducción

A partir del estudio del estado de la cuestión se constata que las medidas de productividad en IS a nivel de puesto de trabajo, en general, se centran en el empleo del tiempo y esfuerzo como unidades de medición de los recursos empleados (entradas) y en el tamaño del producto desarrollado (salidas) mediante una relación de ratio. Además, en los resultados de la SLR se ha encontrado un estudio que plantea la posibilidad de medir la productividad basándose en las tareas realizadas (Hernández-López, et al., 2013). Estos resultados no son concluyentes por lo que es necesario continuar investigando sobre la medición de la productividad a nivel de puesto de trabajo. Además, para contrastar las hipótesis planteadas es necesario disponer de información que emane de los propios trabajadores cuya productividad será medida con las futuras medidas de productividad.

Una forma de resolver este doble problema, por un lado, la obtención de información y, por otro lado, el contraste de hipótesis, es mediante la utilización de metodologías cualitativas cuyo carácter exploratorio permite obtener conocimiento desde los objetos de investigación (en este caso trabajadores de IS), construir teorías a partir de conocimiento, y contrastar las hipótesis planteadas. Además, como complemento a la metodología cualitativa, y para asegurar el contraste de las hipótesis

mediante métodos estadísticos, se plantea la utilización de una metodología cuantitativa.

Las hipótesis que se contrastarán en este enfoque del problema son las siguientes:

- **Hipótesis 1.** En los puestos de trabajo involucrados en la ejecución de proyectos de desarrollo de software se emplean otras entradas, además del tiempo y el esfuerzo.
- **Hipótesis 2.** Las entradas utilizadas son distintas para cada puesto de trabajo involucrado en la ejecución de proyectos de desarrollo de software.
- **Hipótesis 3.** En los puestos de trabajo involucrados en la ejecución de proyectos de desarrollo de software se producen otras salidas, además de líneas de código y la funcionalidad.
- **Hipótesis 4.** Las salidas producidas son distintas para cada puesto de trabajo involucrado en la ejecución de proyectos de desarrollo de software.
- **Hipótesis 5.** Las actuales medidas de productividad a nivel de puesto de trabajo en los proyectos de desarrollo de software tienen una eficacia limitada para medir la productividad real de los trabajadores.

3.2. Justificación y método

Con el objetivo de contrastar las hipótesis planteadas, y teniendo en cuenta el estudio del estado de la cuestión realizado, se ha diseñado un método dividido en dos fases: una primera fase cualitativa y una segunda cuantitativa para: (1) contrastar las hipótesis y (2) obtener información suficiente para poder construir nuevas medidas de productividad a nivel de puesto de trabajo en proyectos de desarrollo de software. Se ha optado por este tipo de diseño ya que la información disponible en la literatura sobre los conceptos bajo investigación no es amplia (Hernández-López, et al., 2013), lo que hace necesario extraer información desde los puestos de trabajo.

El método diseñado para conseguir estos objetivos se ilustra en la Ilustración 9. El objetivo de la fase cualitativa es conocer la realidad desde el punto de vista de los propios trabajadores y obtener información para contrastar con información cualitativa las hipótesis de investigación H1, H2, H3 y H4. Y, el objetivo de la fase cuantitativa es contrastar estadísticamente las hipótesis H1, H2, H3 y H4. Así, en los apartados 3.3 Fase cualitativa y 3.4 Fase cuantitativa se presentan las dos fases, junto con los resultados obtenidos y una breve discusión de los mismos para cada ellas. Finalmente se describe la situación de la tesis tras la realización de ambas fases de estudios en el apartado 3.5 Situación tras ambas fases.

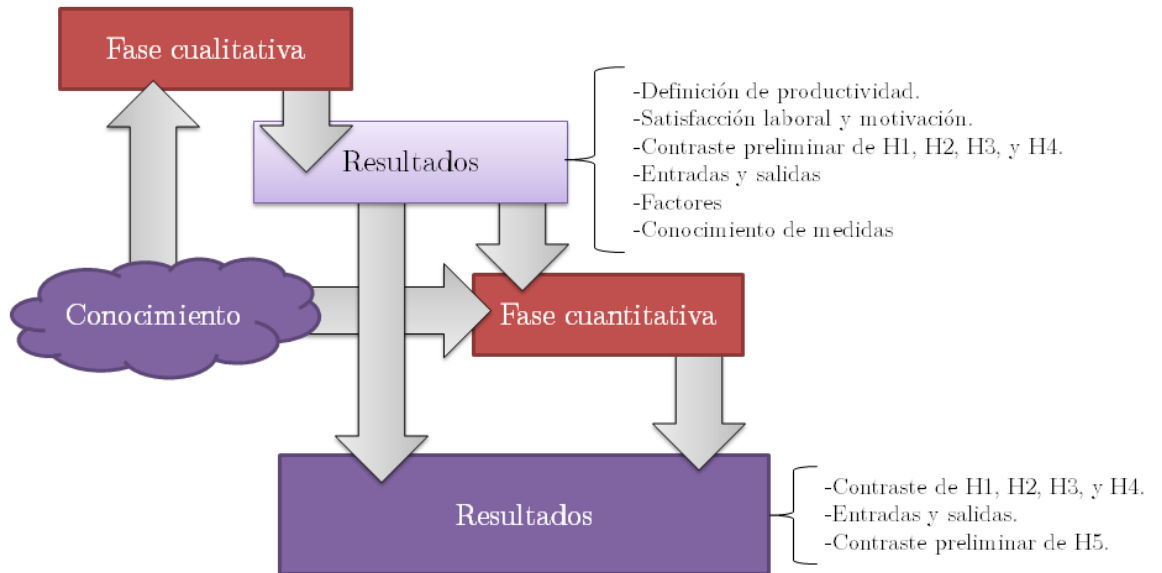


Ilustración 9. Proceso de enfoque del problema

3.3. Fase cualitativa

La investigación cualitativa se ha utilizado principalmente en la investigación de sucesos sociológicos, culturales y antropológicos, es decir, situaciones en las que hay seres humanos involucrados (Denzin & Lincoln, 2011). Dado que las actividades a realizar dentro de los proyectos de desarrollo de software necesitan una fuerte carga de capital humano, utilizar una metodología cualitativa está justificada (Hove & Anda, 2005). Además, estas metodologías investigan desde los datos para crear teorías, lo que

representa un apoyo sustancial una ayuda para la realización de la investigación dado que el objeto de estudio (productividad de los puestos de trabajo vinculados con los proyectos de desarrollo de software) es una área que requiere más investigación (Petersen, 2011; Ramirez & Nembhard, 2004), más aún tras constatar que la investigación es escasa (Hernández-López, et al., 2013). Este tipo de metodología ha sido utilizado en el área de investigación de la medición de la productividad los trabajadores del conocimiento de forma satisfactoria (Erne, 2011). Así pues, con el objetivo de contrastar las cuatro primeras hipótesis planteadas, y para obtener información suficiente para construir posteriormente la fase cuantitativa, se ha diseñado y ejecutado el siguiente método cualitativo.

3.3.1. Metodología

Dada la naturaleza exploratoria de esta fase de la investigación, se ha decidido usar una metodología cualitativa basada en el uso de análisis de contenido a partir de entrevistas semiestructuradas, utilizadas como forma de recogida de información (Kvale, 2008). Esta forma de recolección aporta información que no podría ser obtenida mediante una aproximación cuantitativa (p. ej., basada en formularios) dado que permite a los participantes expresar sus opiniones, pensamientos y sentimientos (H. Rubin & Rubin, 2011). El proceso específico que se ha diseñado y ejecutado está resumido en la Ilustración 10.

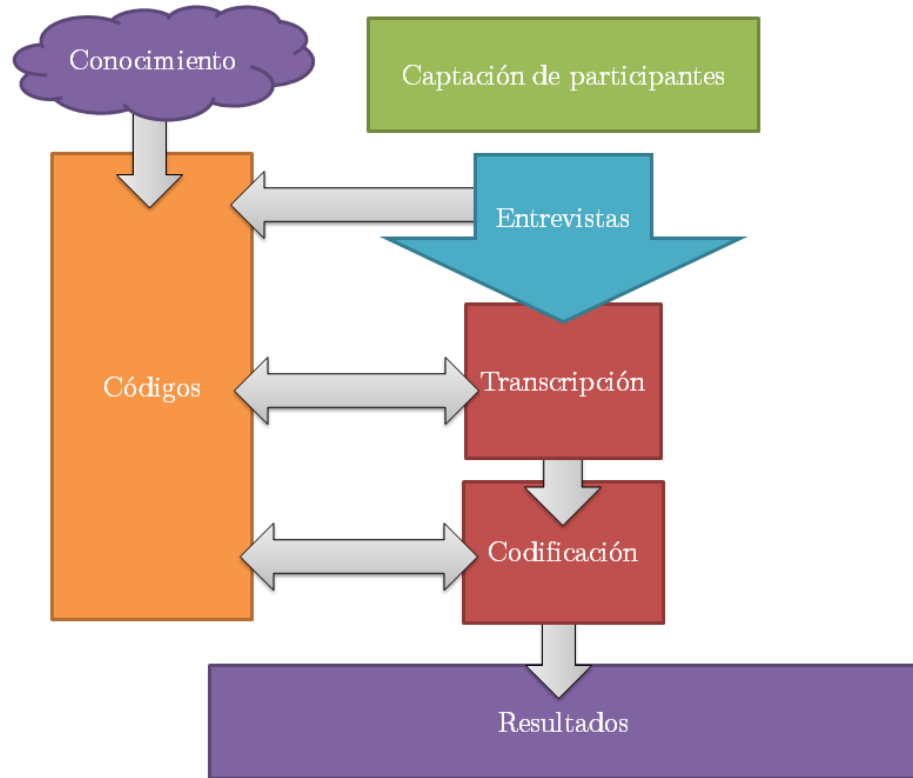


Ilustración 10. Proceso de trabajo de la fase cualitativa

El proceso para obtener los participantes fue el siguiente. En primer lugar, D. Ricardo Colomo Palacios contactó vía email con ex-alumnos con experiencia en actividades de IS, y se estableció un mínimo de 10 participantes para empezar a realizar las entrevistas. De estos contactos, se obtuvieron 15 participantes interesados en participar. Así, dado que el número de interesados era mayor que el número mínimo fijado, se comenzó a realizar las entrevistas; éstas se realizaron entre abril y septiembre de 2011, y se realizaron en español. La muestra final de participantes tuvo las siguientes características: 15 participantes (14 hombres y 1 mujer), de los cuales 4 fueron Jefes de Proyecto (JPs), 11 Ingenieros Software (ISs). La media de edad fue de 30,47 años (5,18 de), con una media de antigüedad en el puesto de 2,56 años en el puesto (2,71 de), y una media de 5,93 años en el campo de la IS (4,85 de), y en concreto 10,75 años de antigüedad en IS en el caso de los JPs y 6,58 los ISs. El tiempo total de entrevistas fue de 9 horas, 43 minutos y 18 segundos, con una media de 38 minutos y 53 segundos por entrevista. Esta muestra tiene una baja representación de

mujeres (1/15), en comparación con la presencia de mujeres en el sector en España que se sitúa en torno al 23% (INE, 2011).

Las entrevistas fueron realizadas por el autor de la tesis doctoral (que, en el momento de realizarlas, poseía conocimiento sobre el área de investigación pero no podría considerarse un experto), fueron grabadas y transcritas por él mismo. Para la transcripción y codificación se empleó el software *Atlas ti 6*. El proceso utilizado para la codificación fue el siguiente: primero se realizó una lista de códigos que cubrían los elementos principales de la investigación a partir del conocimiento general sobre el área de investigación; en segundo lugar, se codificaron las entrevistas y se crearon códigos adicionales para dotar de mayor precisión a la codificación; en tercer lugar, una vez todos los datos fueron codificados, se aislaron los datos relativos a los conceptos bajo estudio: definición de productividad, entradas utilizadas, salidas producidas, motivación y satisfacción laboral, factores que afectan a la productividad, utilidad de las medidas actuales para medir productividad individual, dificultades para medir entradas, y dificultades para medir salidas.

3.3.2. Resultados

A continuación se presentan los resultados de forma separada en cinco apartados: (1) definición de productividad por parte de los participantes en tres niveles: organización, proyecto, e individual, (2) entradas utilizadas y salidas producidas por los participantes en sus puestos de trabajo, (3) satisfacción laboral y motivación, (4) factores que afectan a la productividad, y (5) conocimiento de las medidas de productividad más utilizadas (SLOC/t y PF/t).

3.3.2.1. Definición de productividad

Para obtener una definición de productividad, los datos se han dividido en tres niveles: organización, proyecto, y puesto de trabajo. La definición a nivel de puesto de trabajo se divide en dos grupos dado que los participantes forman parte de dos puestos

de trabajo claramente diferenciados: jefe de proyecto e ingeniero de software. Para cada una de estas definiciones se ha incluido una tabla con los términos mencionados por los participantes y el número de ocurrencias de dicho término (ver desde Tabla 9 a Tabla 11).

Además se han incluido los sub-términos relacionados con los términos y el número de ocurrencias de los mismos. Estos sub-términos tienen mayor grado de detalle y se incluyen para mostrar la variedad de términos mencionados. A partir de estos términos y sub-términos fue posible construir una definición de productividad para cada nivel.

A nivel de organización, los participantes mencionan siete conceptos (entre paréntesis el número de ocurrencias): *tiempo* (7), *producto* (6), *ventas* (5), *cumplimiento del trabajo* (4), *recursos* (4), *calidad* (4), y *objetivos* (3). Y otros términos, con sólo una mención: *necesidades del cliente*, *gestión de riesgos*, y *costes*. La definición a nivel de proyecto se centra en cinco elementos, algunos de ellos también incluidos en las definiciones a nivel de trabajador: *progreso* (8), *tiempo* (6), *calidad* (4), *recursos* (3), *solución de bugs* (3), y *entregables* (3). Además, los participantes señalan otros términos a este nivel de definición: *objetivos* (2), *ventas* (2), y *costes de recursos humanos* (1). En el caso de la definición de productividad para los ISs se mencionan principalmente cinco términos: *tiempo* (11), *cumplimiento del trabajo* (8), *calidad* (6), *solución de bugs* (4) y *solución de problemas* (3). Además, se mencionan la entrega de *funcionalidad* (2) y la *satisfacción del cliente* (1). Para los JPs, la definición es muy dispar y tiene sólo dos términos repetidos: *cumplimiento del trabajo* (2), y *solución de problemas* (2). Todos los demás términos sólo fueron mencionados una vez: *gestión del trabajo*, *gestión de recursos*, *desarrollo de personas*. Hay que tener en cuenta que el *tiempo* fue mencionado tanto por los ISs como los JPs. De forma gráfica se muestran los términos y los niveles en la Ilustración 11.



Ilustración 11. Términos para la definición de productividad

En conjunto hay cuatro términos que se repiten en todas las definiciones: *tiempo*, *cumplimiento del trabajo*, *objetivos*, y *calidad*. Estos cuatro elementos son transversales y deberían estar presentes en todas las definiciones. De este modo, es posible definir la productividad para cada uno de los niveles de esta forma (en cursiva los términos):

- **Definición de productividad de una Organización productora de software:** *Venta de un producto con la calidad requerida, realizando los trabajos necesarios y utilizando los recursos necesarios para que los objetivos se cumplan dentro del tiempo especificado.*
- **Definición de productividad de Proyecto de desarrollo de software:** *Progreso del proyecto, con la calidad requerida y utilizando los recursos necesarios para producir los entregables y resolver los bugs, de modo que se cumplan los objetivos, dentro del tiempo especificado.*

- **Definición de productividad de un Jefe de Proyecto software:**
Cumplimiento del trabajo, solucionando los problemas, gestionando recursos, controlando y mejorando la eficacia y rendimiento de los mismos, dentro del tiempo especificado.
- **Definición de productividad de un Ingeniero de Software:**
Cumplimiento del trabajo, con la calidad requerida, solucionando los problemas y bugs encontrados, dentro del tiempo especificado.

Teniendo en cuenta la clasificación de Ghobadian y Husban (1990), estas definiciones encajan de la siguiente manera: la definición de Ingeniero de Software encaja en la visión ingenieril porque se centra en la relación entre el trabajo realizado y el trabajo planificado; la definición de Jefe de Proyecto encaja también en la clasificación ingenieril pero hay conceptos de la visión económica dado que se mencionan la eficiencia y el rendimiento; la definición de proyecto encaja con los tres tipos posibles dado que tiene elementos de todas ellas; y finalmente, la definición de organización encaja también con todas ellas pero es más global ya que sustituye el concepto de entregables utilizado en el Proyecto por el concepto de producto.

Término	Ocurrencias	Sub-término
Tiempo	11	-
Cumplimiento del trabajo	8	Tarea Objetivo
Calidad	6	Calidad del código fuente Software libre de fallos
Solución de bugs	4	Nuevos bugs Bugs que vuelven
Solución de problemas	3	-
Entrega de funcionalidad	2	-
Satisfacción del cliente	1	-

Tabla 9. Términos de la definición de productividad para Ingeniero de Software

Término	Ocurrencias	Sub-término
Cumplimiento del trabajo	2	Tareas Objetivos
Solución de problemas	2	-
Tiempo	1	-
Gestión del trabajo	1	Tareas Proyecto
Gestión de recursos	1	-
Desarrollo de personas	1	-
Producción	1	Rendimiento Eficiencia
Documentación	1	-

Tabla 10. Términos de la definición de productividad para Jefe de Proyecto

Término	Ocurrencias	Sub-término
Progreso	8	Hitos Tareas
Tiempo	6	-
Calidad	4	Mantenibilidad Rendimiento
Recursos	3	Recursos material RR.HH.
Solución de bugs	3	Nuevos bugs Bugs que vuelven
Entregables	3	-
Objetivos	2	Resultados
Ventas	2	-
Coste de RR.HH.	1	-

Tabla 11. Términos de la definición de productividad para Proyecto

Término	Ocurrencias	Sub-término
Tiempo	7	-
Producto	6	Entregable Cantidad
Ventas	5	-

Término	Ocurrencias	Sub-término
Cumplimiento del trabajo	4	-
Recursos	4	Recursos materiales RR.HH.
Calidad	4	-
Objetivos	3	Resultados
Necesidad del cliente	1	-
Gestión de riesgos	1	-
Costes	1	-

Tabla 12. Términos de la definición de productividad para Organización

Teniendo en cuenta estos resultados, se analizaron las respuestas específicas sobre dos asuntos relacionados con la gestión del tiempo y del trabajo. Por un lado, si los participantes computan horas de trabajo, y, en caso de hacerlo, a qué concepto vinculan dichas horas. Por otro lado, se analizan las respuestas a la pregunta de si consideran útiles las medidas basadas en SLOC y PF en caso de conocerlas para medir la productividad de su trabajo.

Con respecto a las horas computadas, sobre tres cuartas partes de los participantes (11/15) computan horas mediante algún sistema: desde simples hojas de cálculo a sistemas software específicos, y entorno a uno de cada cuatro (4/15) no computan horas de ninguna manera. De los participantes que computan horas, algo menos de la mitad (5/11) asignan a nivel de tarea, y algo más de la mitad (6/11) a nivel de proyecto. Algunos de los que asignan a tareas también asignan a nivel de proyecto (3/5), y a tareas específicas como resolución de bugs (1). Un participante menciona que ellos tienen una clasificación de tareas: cliente, formación, trabajo departamental, vacaciones, y enfermedad. Además, varios participantes indicaron que las organizaciones clientes suelen requerir que se computen horas en un sistema paralelo, estos mismos participantes señalan que esos sistemas son más específicos en cuanto a la granularidad de las tareas. Además de estas características, varios participantes comentan que la justificación para computar horas de trabajo se debe exclusivamente

a la subcontratación y a la justificación de la presencia del trabajador en un determinado proyecto.

Los participantes incluyen el cumplimiento del trabajo dentro de la definición a nivel de proyecto mediante la inclusión del concepto de progreso. Este concepto, junto con el de tarea, son elementos clave en la medición de la productividad de acuerdo a diversos investigadores. Por ejemplo, Mundel (1975) indica que es posible medir la productividad de los trabajadores del conocimiento si hay respuesta a las siguientes preguntas: cuál es el objetivo de la tarea, cuales son las salidas que tienen que ser producidas para que el trabajo se complete, cómo se pueden contar estas salidas, cuánto y qué tipo de recursos son necesarios para producir las salidas. Mediante la respuesta a estas tareas, Mundel indica que se obtiene un punto de partida para medir la productividad. Algunas de estas preguntas fueron incluidas en las entrevistas realizadas para esta fase de la investigación; para más información sobre las preguntas realizadas a los participantes ver el Anexo C - Guión de entrevista semiestructurada.

Por otro lado, casi todos los participantes conocen las medidas basadas en SLOC o PF para medir la productividad (13/15). De estos participantes, todos ponen énfasis en asegurar que de entre estas dos medidas, las basadas en SLOC no poseen validez para medir la productividad a nivel de trabajador, y en torno a la mitad (7/15) descartan ambas para medir su productividad. Además, uno de cada tres participantes (5/15) indican que para medir su productividad con una medida basada en PF sería necesario realizar algún tipo de ajuste y no tienen claro si directamente podría realizarse. Un solo participante señala que las medidas basadas en PF pueden ser útiles para medir su productividad. Estos resultados indican un total desacuerdo con el uso de SLOC como medida de salida para medir la productividad y abre la posibilidad de medir la productividad de los trabajadores a través de los PF; esta posibilidad ya ha sido discutida en un artículo centrado en la medición de la productividad de los trabajadores del conocimiento (Ramirez & Nemhard, 2004).

Las definiciones obtenidas en esta fase de la investigación son más abstractas que la que aporta por ejemplo la norma IEEE Std. 1045-1992 que define la productividad como la relación de una primitiva de salida (p. ej., líneas de código, puntos función o documentos) y su correspondiente primitiva de entrada (p. ej., esfuerzo) para desarrollar software. Además, estas definiciones son más abstractas que las definiciones usadas como SLOC/t o PF/t. No obstante, las citadas definiciones tienen un punto en común con las definiciones obtenidas: el uso del tiempo como recurso; pero se diferencian en el uso del concepto de trabajo en sustitución de salidas específicas tales como las SLOC o PF.

3.3.2.1.1. Observaciones

Un aspecto importante a tener en cuenta es que la mayoría de los participantes computan horas a un concepto más general (proyecto) que el empleado en la definición obtenida de productividad a nivel de trabajador (tareas). Este aspecto crea una controversia dado que si se quiere medir la productividad de forma ajustada a la definición obtenida será imposible hacerlo porque las horas registradas no están vinculadas a las tareas realizadas.

Otro aspecto importante a tener en cuenta es la inclusión del concepto de la solución de bugs tanto a nivel de trabajador como de proyecto. Este concepto no se incluye en las medidas más utilizadas de productividad en IS a pesar de que diversos estudios comparan estos conceptos (ver p. ej.: Maurer & Martel, 2002; Phipps, 1999), y en algunas medidas de productividad se incluye como código fuente con bugs (W. D. Yu, et al., 1991). En esta línea, varios de los participantes comentaron que las organizaciones en las que actualmente trabajan, miden la productividad en parte por el número de bugs abiertos y reabiertos. Esos mismos participantes tienen establecidos determinados días de la semana dedicados a la solución de bugs, siendo una tarea específica en su planificación.

Además, otro concepto que aparece relacionado con la definición es la calidad, que está presente en los tres niveles de definición analizados. Este concepto ha sido estudiado en múltiples ocasiones en relación con la productividad (ver p. ej.: Krishnan, et al., 2000). A nivel de proyecto, dos participantes mencionaron la mantenibilidad y la eficiencia como elementos que definen la productividad a este nivel. Estos factores están incluidos en la norma ISO/IEC 25000:2005. Igualmente, dado que los participantes centran la definición en el concepto de tarea, se podría considerar la posibilidad de utilizar medidas de calidad de tarea y no sólo considerar la calidad del código o los entregables. De hecho, esto se relaciona con la importancia de la calidad en detrimento de la cantidad en los trabajadores del conocimiento tal y como Drucker menciona (1999), o la visión de la calidad como una dimensión de la productividad de este tipo de trabajadores (Ramirez & Nembhard, 2004).

Adicionalmente, los conceptos económicos aparecen en los niveles superiores de medición. Por ejemplo, los participantes mencionan las ventas (2) y los costes de recursos humanos (1) a nivel de proyecto, y las ventas (5) y costes (1) a nivel de organización. En esta línea, los recursos fueron mencionados tres veces a nivel de proyecto y cuatro a nivel de organización, pero a nivel de trabajador el único recurso mencionado es el tiempo (del trabajador). Sin embargo, en otras partes de las entrevistas los participantes mencionaron otros recursos pero no los incluyeron en su definición de productividad.

A pesar de que no es posible generalizar a partir de los resultados obtenidos, la satisfacción del cliente podría considerarse parte de la definición de productividad. Otras dimensiones tales como la innovación y creatividad, autonomía, y cantidad fueron mencionadas de forma indirecta. Con respecto a la creatividad, es posible considerarla como una forma de resolver los problemas encontrados (término que si se menciona en las definiciones con más frecuencia). La cantidad, como concepto, no fue incluida en las definiciones dado que sólo fue mencionado de forma directa por dos ISs;

en concreto mencionaron la cantidad de funcionalidad entregada, y un JP que mencionó la cantidad de documentación generada.

Finalmente, teniendo en cuenta las dimensiones utilizadas para medir la productividad (Ramirez & Nembhard, 2004), las definiciones obtenidas encajan en varias de estas dimensiones. En concreto, las definiciones obtenidas encajan en las siguientes dimensiones:

- **Organización:** cantidad, coste, oportunidad (tiempo), eficiencia, efectividad.
- **Proyecto:** cantidad, coste, oportunidad (tiempo) eficiencia, efectividad, éxito del proyecto.
- **Jefe de proyecto:** oportunidad (tiempo), eficiencia, efectividad, éxito del proyecto.
- **Ingeniero de software:** oportunidad (tiempo), eficiencia, calidad, eficiencia.

3.3.2.1.2. Discusión

Mediante el uso de una metodología cualitativa ha sido posible obtener una definición de productividad para tres niveles de medición: trabajadores (ingeniero software y jefe de proyecto), proyecto, y organización. Estas definiciones deben ser consideradas como exploratorias, es decir, como un punto de partida que tiene que ser tratado en los siguientes pasos de la investigación. Sin embargo, pese a esta limitación, es posible obtener algunas conclusiones, aunque puedan resultar obvias.

La visión que los ingenieros software participantes señalan es más abstracta que la que se tiene en cuenta para crear las medidas actuales de productividad a nivel de trabajador. En concreto se abstrae el concepto de desarrollo y la productividad se centra a nivel de tareas, sean o no de desarrollo. Así, utilizar medidas específicas tales como SLOC/t o PF/t, para medir la productividad de estos trabajadores, tiene una

validez limitada a las tareas exclusivamente de desarrollo, por lo que su eficacia para un propósito más general debería ser evaluada.

A medida que el nivel de medición incrementa, la granularidad a emplear en la medición de las salidas aumenta. Por ello, a nivel de trabajador, las salidas se centran en las tareas y el trabajo, a nivel de proyecto en el proyecto, y a nivel de organización en el producto. Este hecho también sucede en el caso de las entradas, pero una de ellas permanece en todos los niveles: el tiempo. Así pues, la granularidad de las medidas debería ser reducida o aumentada a medida que se aumente o reduzca el nivel de medición. No obstante, las definiciones obtenidas están centradas en el éxito del proyecto: a nivel de trabajador centrado en las tareas y a nivel de organización en el producto. De este modo, el concepto de productividad se vincula con la estimación y planificación de los proyectos.

Otra conclusión que se puede obtener es la relación entre la calidad y la productividad mediante la inclusión de la misma en la definición de productividad en todos los niveles. Este concepto, no está incluido en las medidas de productividad tradicionalmente utilizadas a pesar de ser considerado como una salida importante cuando se mide la productividad en IS. En esta dirección, se menciona la solución de bugs en las definiciones a nivel de trabajador y de proyecto aunque tampoco está incluido en la mayoría de las medidas. Estas exclusiones son realmente importantes dado que las actividades de solución de bugs y mantenimiento consumen una gran proporción de recursos y esfuerzo en el éxito del proyecto. Así pues, ¿debería considerarse como igualmente productivo un proyecto a pesar de los recursos destinados a la solución de bugs y mantenimiento correctivo? A priori puede decirse que las medidas tradicionalmente utilizadas (p. ej., SLOC/t o PF/t) no resuelven esta cuestión. Así pues, la inclusión de la calidad debería ser considerada cuando se desarrollen nuevas medidas de productividad para IS, en cualquier nivel de medición.

Estas conclusiones no deben ser consideradas como probatorias teniendo en cuenta la naturaleza exploratoria del método empleado y la reducida muestra utilizada.

Además, la muestra incluye participantes con diferente antigüedad en IS y no se puede considerar una muestra madura en el sector: la edad media es de 30,47 años, la antigüedad media en el trabajo 2,56 años, y la media en IS de 5,93 años.

3.3.2.2. Entradas y salidas

Antes de transcribir las entrevistas, se ha elaborado una lista inicial de códigos que incluye las entradas y salidas que pueden haberse mencionadas. Estos códigos se basan en el conocimiento del área de estudio. Durante la transcripción y codificación, se han añadido códigos adicionales para especificar con mayor grado de detalle los elementos mencionados por los participantes. Finalmente, se ha creado una lista de códigos con subcódigos más específicos. A los elementos marcados como códigos se les denomina ítem y a los marcados con subcódigos sub ítem. Por ejemplo, hay un término marcado como *calidad* con varios subtérminos llamados *calidad de código fuente*, *calidad de producto*, y *calidad añadida*. A continuación se presentan los resultados para las entradas y salidas de forma separada.

3.3.2.2.1. Entradas

Las entradas mencionadas por los participantes están incluidas en la Tabla 13. La entrada más mencionada fue el *tiempo* (n=14) incluyendo: *fechas* (8), *planificación* (5), y *entrega en tiempo* (3). Este resultado está en consonancia con la importancia del tiempo en las medidas más usadas en la medición de la productividad. Además, la *gestión del trabajo* fue mencionada por casi todos los participantes (14) junto con varios conceptos relacionados con el mismo: *asignación de objetivos* (6), *asignación de tareas* (6), *estimación* (4). Con menos menciones, hay otras entradas: *requisitos* (11), *conocimiento* (11), *cliente* (11), *recursos* (13), *documentos* (8), *experiencia* (8), y *educación y formación* (3). Además hay otras de carácter puramente intangible (9) como las *competencias de gestión* (2), *colaboración de equipo* (1), y otras de carácter más tangible como el *código fuente* (2) y los *casos de soporte* (1).

Dentro de los requisitos, que es una entrada utilizada en diversas tareas en IS, su definición es la más citada (10) junto con los cambios en ellos (5). Además, la *integración del cliente para la toma de requisitos* (7), la cual está incluida dentro de la entrada cliente, está relacionada con los *requisitos*. De forma adicional, los *gustos del cliente* (1), que está incluida dentro de los elementos intangibles, está relacionada con la entrada *requisitos*. Así pues, existe una relación entre los requisitos y el cliente a la luz de estos resultados.

Con respecto a las demás entradas, se mencionan los siguientes elementos más de una vez: *software* (8), *hardware* (7), *documentación en el software* (5), *experiencia propia* (5), *documentación del proyecto* (4), *experiencia del equipo* (3), *formación específica* (3), *experiencia en tareas similares* (3), *instalaciones de trabajo* (2), *formación continua* (2), *recursos materiales* (2), *know-how* (5), *resolución de dudas* (3), *conocimiento del proyecto* (3), *conocimiento funcional* (2), y *producto previo* (2).

Ítem	N (ISs+JPs)	Sub Ítem	N
Tiempo	14 (11+3)	Tiempo	14
		Fechas	8
		Planificación	5
		Entrega en tiempo	3
Gestión del trabajo	14 (10+4)	Planificación	9
		Asignación de objetivos	6
		Asignación de tareas	6
		Estimación	4
		Reuniones de seguimiento	4
		Comunicación con el equipo	2
		Modelos de gestión	2
		Procesos de trabajo	1
Recursos	13 (10+3)	Software	8
		Ordenador (y Hardware)	7
		Instalaciones de trabajo	2
		Recursos materiales	2
		Puesto de trabajo	1

Ítem	N (ISs+JPs)	Sub Ítem	N
		Teléfono	1
Requisitos	11 (9+2)	Definición de requisitos	10
		Cambios en los requisitos	5
		Cumplimiento de requisitos	1
Conocimiento	11 (8+3)	Conocimiento	8
		Know-how	5
		Resolución de dudas	3
		Conocimiento del proyecto	3
		Conocimiento funcional	2
Cliente	11 (8+3)	Integración para la toma de requisitos	7
		Frecuencia de interacción con el cliente	4
		Interacción para los test de aceptación	2
		Interacción constante con el cliente	1
Otras intangibles	9 (7+2)	Competencias de gestión de equipo	2
		Producto previo	2
		Colaboración con el equipo	1
		Yo mismo	1
		Recursos humanos	1
		Información	1
		Dificultad de la tarea	1
		Diseño	1
		Gustos del cliente	1
Documentación	8 (7+1)	Documentación de software	5
		Documentación de proyecto	4
		Informes	1
		Email	1
		Estándar	1
Experiencia	8 (4+4)	Experiencia propia	5
		Experiencia del equipo	3
		Experiencia en tareas similares	3
Otras tangibles	5 (5+0)	Código fuente previo	2
		Casos de reporte	1
Educación y formación	3 (2+1)	Formación específica	3
		Formación continua	2

Ítem	N (ISs+JPs)	Sub Ítem	N
		Formación universitaria	1

Tabla 13. Entradas mencionadas por los participantes

3.3.2.2.2. Salidas

Las salidas mencionadas por los participantes están incluidas en la Tabla 14. La salida más mencionada es la *documentación* (n=13), con varias salidas de más detalle como *documentación de diseño* (6), *análisis* (6), *proyecto* (4), *proceso* (3), *código fuente* (2), y *otra tipo de documentación no especificada* (7). Seguida esta salida en términos de menciones recibidas, los participantes mencionan el *código fuente* (10), *gestión del trabajo* (9), *conocimiento* (9), *calidad* (8), *ventas* (7), *pruebas* (5), y *experiencia* (1). Los participantes a su vez mencionan otras salidas tangibles (12) como *producto* (9) y *especificación de requisitos* (2); y otras intangibles (12) como *solución de problemas y código fuente* (6), *análisis* (4), *proyecto* (3), *funcionalidad* (3), *diseño* (3), *satisfacción del cliente* (3), y otras salidas con una sola mención.

Las dos salidas más utilizadas en las medidas de productividad están presentes en los resultados obtenidos: *código fuente* (10) y *funcionalidad* (3). Además, la *documentación* que suele ser medida en función del tamaño del código fuente está presente en los resultados (13) siendo la más mencionada de las salidas.

De estos resultados es importante señalar que varios elementos fueron mencionados como salidas y como entradas. Por ejemplo, el *conocimiento* fue mencionado tanto como salida (9) como entrada (11), y lo mismo sucede con la *experiencia*, salida (9) y entrada (1). Por lo tanto, a partir de los resultados se puede decir que existe un proceso de transformación de varias entradas en salidas durante la ejecución de proyectos software. Este hecho añade una dificultad añadida en la tarea de medición de la productividad de los puestos de trabajo vinculados a dichas tareas.

Ítem	N (SE+PM)	Sub Ítem	N
Documentación	13 (10+3)	Documentación	7

Ítem	N (SE+PM)	Sub Ítem	N
		Diseño	6
		Análisis	6
		Proyecto	4
		Proceso	3
		Código fuente	2
Otras tangibles	12 (9+3)	Producto	9
		Calidad	2
		Especificación de requisitos	2
Otras intangibles	12 (8+4)	Solución de bugs y problemas	6
		Análisis	4
		Proyecto	3
		Funcionalidad	3
		Diseño	3
		Satisfacción del cliente	3
		Elemento abstracto	1
		Arquitectura	1
		Experimentación	1
		Puntos (Historias de usuario)	1
		Gestión de equipo	1
		Mejora y mantenimiento	1
		Cumplimiento de requisitos	1
Código fuente	10 (7+3)	Corrección de código fuente	2
Gestión de trabajo	9 (7+2)	Hacer y finalizar tareas	7
		Estimación	4
		Planificación	3
		Cumplimiento de objetivos	2
		Asignación de tareas	2
		Coordinación	2
		Trazabilidad	1
Conocimiento	9 (8+1)	Aprendizaje en el puesto	3
		Soluciones encontradas	2
		Resolución de dudas	2
		Información	1
		Conocimiento incluido en la documentación	1

Ítem	N (SE+PM)	Sub Ítem	N
		Conocimiento del proyecto	1
Calidad	8 (6+2)	Calidad (en general)	6
		Calidad del código fuente	3
		Calidad del producto	2
		Calidad del desarrollo de personas	1
		Rendimiento del software desarrollado	1
		Calidad añadida	1
Ventas	7 (4+3)	Ventas	7
Pruebas	5 (4+1)	Pruebas de software	5
		Casos especiales	1
Experiencia	1 (1+0)	Experiencia	1

Tabla 14. Salidas mencionadas por los participantes

3.3.2.2.3. Disucusión

La entrada más mencionada por los participantes es el *tiempo* incluyendo otros elementos de mayor detalle como *fechas*, *planificación*, y *entrega en tiempo*. Este resultado está en línea con la gran visibilidad que tiene este elemento en las medidas de productividad en cualquier nivel de medición. Sin duda es un recurso necesario ya que sin tiempo es imposible desempeñar ninguna actividad, y por consiguiente debe estar presente en la medición de la productividad. No obstante, se mencionaron otras entradas que no pueden ser medidas utilizando el tiempo como elemento *proxy*. Por ejemplo, los elementos relacionados con la *gestión del trabajo* como la *asignación de objetivos* y *de tareas*, y la *estimación*, son elementos atemporales para el desempeño de las funciones de los trabajadores, es decir, aunque el tiempo pase estos elementos estarán presentes en su trabajo y no serán objeto de constante cambio con el paso del tiempo. Además, estos elementos junto con otros tales como los *requisitos*, el *código fuente previo*, los *casos de soporte*, entre otros, son elementos generados con anterioridad por otros trabajadores, por lo que también se trata de recursos atemporales. Los participantes también mencionan elementos intangibles como entradas: la *experiencia*, la *colaboración con el equipo*, las *competencias de gestión*, y

la *educación y formación*, entre otras. Así pues, estos resultados indican la existencia de otras entradas, además del *tiempo* y el *esfuerzo*, de diversa naturaleza y tipología. Es decir, la H1 puede considerarse confirmada en base a la información cualitativa disponible.

Por otro lado, la salida más mencionada es la *documentación (diseño, análisis, proyecto, código fuente...)* seguida del *código fuente*. Además, se mencionan otras salidas (algunas de ellas de naturaleza intangible y de difícil medición) como la *gestión del trabajo*, el *conocimiento*, la *calidad*, las *ventas*, la *especificación de requisitos*, la *solución de problemas...* También se menciona la *funcionalidad*, aunque en menor medida. Así pues, de igual forma que para las entradas es posible confirmar la H1, para las salidas es posible hacerlo con la H3 ya que los resultados reflejan la existencia de otras salidas además del código fuente y la funcionalidad.

Además, en los resultados obtenidos está presente el proceso de transformación de varias entradas en salidas durante la ejecución de proyectos software. Por ejemplo, la *especificación de requisitos*, es generada como salida por algún puesto de trabajo y posteriormente utilizada como entrada en ese mismo u otro puesto de trabajo. Esto, además de añadir dificultad a la medición de la productividad, refleja la existencia de un proceso de trabajo con múltiples entradas y salidas, que además son utilizadas de forma interna para producir las salidas finales.

Finalmente hay que destacar que no ha sido posible realizar un análisis de las diferencias de utilización de entradas y de producción de salidas entre los puestos de trabajo analizados en esta fase del estudio. Esto se debe a que las respuestas fueron facilitadas por los participantes de forma abierta, es decir, la falta de mención de un elemento no se puede asimilar a la no existencia de dicho elemento. Así pues, será necesario contrastar las hipótesis H2 y H4 mediante el estudio cuantitativo posterior. No obstante, en los resultados se obtienen datos que apuntan hacia esa posible diferencia. Por ejemplo, en el caso de las *ventas* parece existir cierta diferencia ya fue mencionado por el 75% de los JP y entorno a un tercio de los IS lo mencionan

(36.36%). Sin embargo, tal y como se ha dicho, no es posible dar una respuesta firme a estas cuestiones debido al método de recogida de información utilizado.

3.3.2.3. Satisfacción laboral y motivación

Antes de analizar los resultados, hay que señalar que para un participante esta parte del estudio cualitativo no ha podido ser analizada ya que, por fallos de grabación la entrevista, no ha sido posible recoger la parte relativa a estos elementos, en concreto se corto la grabación momentos antes de tratar estos temas. Como se aprecia en el Anexo C - Guión de entrevista semiestructurada, estos temas se encuentran en el final de la entrevista. Así pues, para esta parte de la investigación se trabaja con 14 participantes, en lugar de 15.

Desde un punto de vista cualitativo, varios elementos relacionados con la satisfacción laboral y la motivación fueron mencionados por los participantes (para conocer todos los elementos ver la Tabla 16, la cual incluye tres categorías: satisfacción con la organización del trabajo, con las condiciones de trabajo y con el tipo de trabajo). Los elementos más mencionados como factores de satisfacción ($n > 2$) fueron el *trabajo en equipo* (14), el *entorno de trabajo* (7), la *colaboración* (6), el *salario* (4), la *confianza* (3), los *problemas entre el jefe y los trabajadores* (3), la *variedad de tareas* (3), la *buena gestión* (3), y *que el trabajo guste* (3).

Desde un punto de vista cuantitativo, los participantes muestran una total satisfacción con sus compañeros de trabajo y con los trabajadores que dependen de ellos (en caso de tenerlos, ya que sólo 4 participantes encajan en esta categoría). La satisfacción con el resto de elementos no es uniforme. Así, el 64,29% está satisfecho con el tipo de trabajo que realiza, el 28,57% ni satisfecho ni insatisfecho y el 7,14% está insatisfecho. Si se ponderan estas tres categorías con un 10 para el caso de satisfecho, 5 para ni satisfecho ni insatisfecho, y 0 para insatisfecho, se obtiene un resultado de 7,86 puntos. Este valor es similar al obtenido para la satisfacción con los jefes y superiores: el 71,43% está satisfecho y el 14,29% está insatisfecho. Utilizando la

escala anteriormente mencionada, se obtiene el mismo valor 7,86 puntos para la satisfacción con los jefes y superiores. Finalmente, el 42,86% está satisfecho con la organización del trabajo, el 21,43% no está satisfecho ni insatisfecho, y el 35,71% no está satisfecho. En este caso, la satisfacción con la organización del trabajo es menor que las otras categorías, con un resultado de 6,07 puntos.

Pregunta	N	Sí	Ni satisfecho ni insatisfecho	No
¿Está satisfecho con el tipo de trabajo que realiza actualmente?	14	64,29%	7,14%	28,57%
¿Está satisfecho con los empleados que dependen de usted?	4	100%	0	0
¿Está satisfecho con sus jefes y responsables?	14	71,43%	14,29%	14,29%
¿Está satisfecho con sus compañeros de trabajo?	14	100%	0	0
¿Está satisfecho con la organización del trabajo?	14	42,86%	21,43%	35,71%

Tabla 15. Satisfacción laboral

Categoría	Ítem	N	Sub-Ítems	N
Organización del trabajo	Equipo	14	Equipo	14
			Colaboración	6
			Equipo completo	1
	Jefes	14	Confianza	3
			Problemas	3
			Mala gestión	1
			Cercanía	1
			Comunicación activa	1
			Sinceridad	1
			Mala relación con los jefes de la organización empleadora	1
	Trabajo	13	Variedad	3
			Gestión	3
			Me gusta	3
			Interesante	2

Categoría	Ítem	N	Sub-Ítems	N
			Autonomía	2
			Centrado en un rol específico	1
			Formación en el trabajo	1
	Entorno	7	Entorno	7
	Comunicación	5	Coordinación entre departamentos y proyectos	1
			Requisitos y necesidades	1
			Comunicación con jefes	1
	Formación	2	Más formación	2
			Formación planificada	1
Condiciones de trabajo	Salario	4	Más salario	4
	Condiciones de trabajo	4	Tipo de contrato	2
			Beneficios sociales	1
			Teletrabajo	1
	Horario	3	Flexibilidad	2
			Horario reducido	1
			Ajustarse al horario	1
	Instalaciones	2	Instalaciones	2
			Cercanía del lugar de trabajo a domicilio	2
Trabajo	Tareas	10	Cómo se asigna el trabajo	2
			Estimación	2
			Proyecto grande	2
			Falta de visión global	1
			Interés	1
			Dificultades	1
			Metodologías	1
	Cliente	2	Cesión ilegal de trabajador	1

Tabla 16. Factores relacionados con la satisfacción laboral

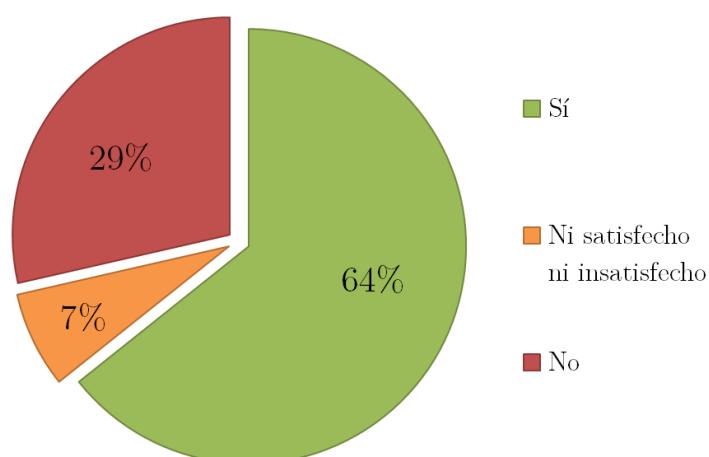


Ilustración 12. Satisfacción con el tipo de trabajo

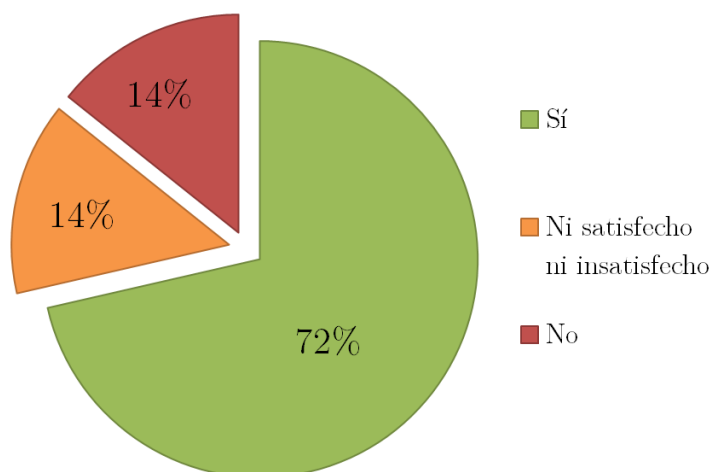


Ilustración 13. Satisfacción con jefes y superiores

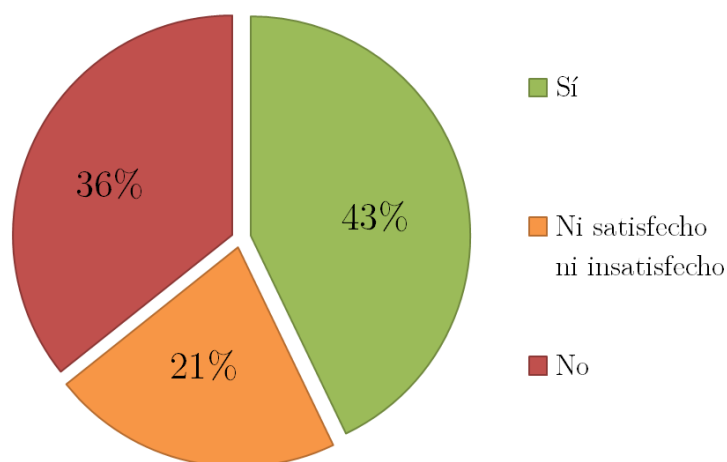


Ilustración 14. Satisfacción con la organización del trabajo

Los participantes mencionan varios ítems que mejorarían su satisfacción laboral (Tabla 17), su productividad (Tabla 20) y su motivación (Tabla 18 y Tabla 19). En la Tabla 17 se han añadido dos columnas para apoyar los resultados con las referencias encontradas en la revisión sistemática sobre motivación en IS realizada recientemente (Beecham, et al., 2008); y en la Tabla 20, se ha incluido una columna con algunas de las referencias que tratan los vínculos entre productividad y los elementos mencionados por los participantes. Los resultados para la satisfacción laboral y la motivación se apoyan gráficamente con la pirámide de necesidades de Maslow (1943) y la teoría de factor dual de Herzberg (1959).

Con respecto a la satisfacción laboral, casi la mitad de los participantes (6) mencionaron el *salario* como una fuente de mejora. También mencionaron que el *trabajo fuera menos monótono* (3), la *formación continua* (2), un *horario de trabajo reducido* (2), y otros elementos con sólo una mención. Estos resultados se muestran gráficamente en la Ilustración 15; ver la lista completa de ítems en la Tabla 17.

Ítem	N	Frecuencia y nombre utilizado en (Beecham, et al., 2008)	Referencias en (Beecham, et al., 2008)
Salario	6	14 - Reward and incentives	(Agarwal & Ferratt, 1998, 2001, 2002; Baddoo, Hall, & Jagielska, 2006; Burn, Couger, Ma, & Tompkins, 1991; Frangos, 1997; Garden, 1988; M. K. Hsu, Chen, Jiang, & Klein, 2003; Klenke & Kievit, 1992; Mak & Sockel, 2001; Niederman & Sumner, 2001; Peters, 2003; Ridings & Eder, 1999; Thatcher, Liu, & Stepina, 2002)

Ítem	N	Frecuencia y nombre utilizado en (Beecham, et al., 2008)	Referencias en (Beecham, et al., 2008)
Menos monótono	3	14 - Variety of Work	(Agarwal & Ferratt, 1998, 2001; Burn, et al., 1991; Garden, 1988; Igbaria, Meredith, & Smith, 1995; Jordan & Whiteley, 1994; Li, Tan, Teo, & Mattar, 2006; Linberg, 1999; Mak & Sockel, 2001; Niederman & Sumner, 2003; Niederman & Sumner, 2001; Ridings & Eder, 1999; Smits, McLean, & Tanner, 1992; Zawacki, 1992)
Formación continua / Reciclaje	2	11 - Development needs addressed	(Agarwal & Ferratt, 1998, 2001; Crepeau, Crook, Goslar, & McMurtrey, 1992; Frangos, 1997; Garden, 1988; Jordan & Whiteley, 1994; Klenke & Kievit, 1992; LeDuc, 1980; Niederman & Sumner, 2003; Santana & Robey, 1995; Tanner, 2003)
Horario reducido	2	7 - Work/life balance	(Agarwal & Ferratt, 1998, 2001, 2002; Garza, Lunce, & Maniam, 2003; Jordan & Whiteley, 1994; Morales, 2005; Richens, 1998)
Más centrado en tareas de gestión y más responsabilidad	2	6 - Empowerment/responsibility	(Agarwal & Ferratt, 1998; Frangos, 1997; LeDuc, 1980; Li, et al., 2006; Smits, et al., 1992)
Cercanía del trabajo a casa	2	-	-
Carrera profesional	1	15	(Agarwal & Ferratt, 1998, 2001; Burn, et al., 1991; Crepeau, et al., 1992; Garden, 1988; Igbaria, et al., 1995; Jordan & Whiteley, 1994; Li, et al., 2006; Mak & Sockel, 2001; Niederman & Sumner, 2003; Niederman & Sumner, 2001; Ridings & Eder, 1999; Smits, et al., 1992; Zawacki, 1992)

Ítem	N	Frecuencia y nombre utilizado en (Beecham, et al., 2008)	Referencias en (Beecham, et al., 2008)
Ser contratado por el cliente, no por la proveedora	1	-	-
Supervisar el impacto de los cambios solicitados por el cliente	1	-	-
Horario de trabajo que se ajuste al horario planificado	1	-	-
Más comunicación con jefes y responsables de la organización empleadora	1	16 - Good Management	(Agarwal & Ferratt, 1998; Baddoo, et al., 2006; Frangos, 1997; Garden, 1988; Jordan & Whiteley, 1994; Klenke & Kievit, 1992; LeDuc, 1980; Li, et al., 2006; Linberg, 1999; Mata-Toledo & Unger, 1985; Niederman & Sumner, 2001; Santana & Robey, 1995; Smits, McLean, & Tanner, 1997; Tanner, 2003; Thatcher, et al., 2002; Zawacki, 1992)
Más proyectos de un tipo específico	1	-	-
Mejor toma de requisitos	1	-	-
Mejor contrato	1	-	-
Mejores instalaciones de trabajo	1	6 - Appropriate working conditions/environment/good equipment/tools/physical space/quiet	(Burn, et al., 1991; Frangos, 1997; Gambill, Clark, & Wilkes, 2000; Garza, et al., 2003; Richens, 1998; H. I. Rubin & Hernandez, 1988)
Metodología	1	-	-

Ítem	N	Frecuencia y nombre utilizado en (Beecham, et al., 2008)	Referencias en (Beecham, et al., 2008)
Reconocimiento	1	12 - Recognition	(Baddoo, et al., 2006; Ferratt & Short, 1986; Frangos, 1997; Garden, 1988; Jordan & Whiteley, 1994; Klenke & Kievit, 1992; LeDuc, 1980; Linberg, 1999; Niederman & Sumner, 2001; Peters, 2003; Procaccino, et al., 2005; Tanner, 2003)

Tabla 17. Elementos que mejoran la satisfacción laboral

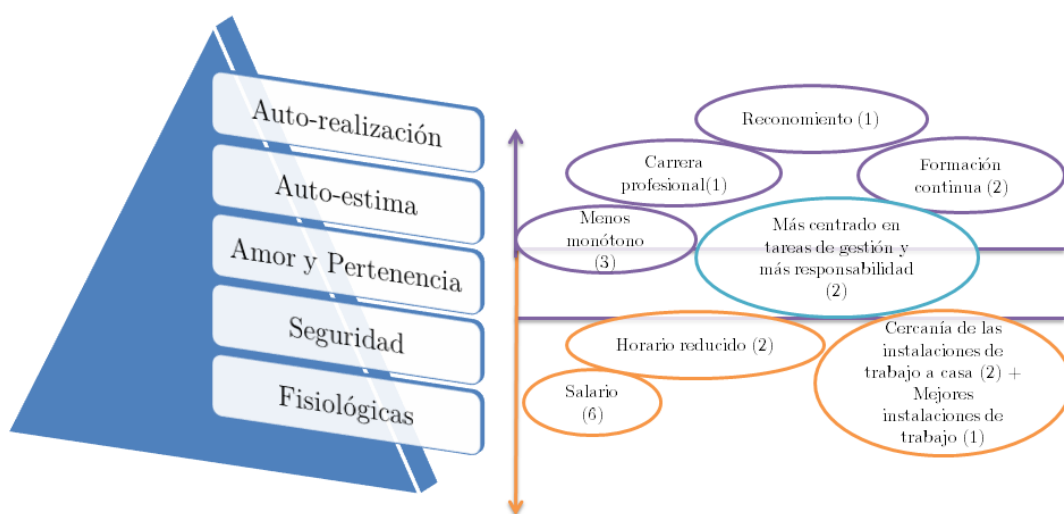


Ilustración 15. Mejora de la satisfacción laboral

Con respecto a la motivación, algunos elementos fueron mencionados como motivadores (Tabla 18) y otros como desmotivadores (Tabla 19). Como elementos motivadores, los participantes mencionaron el *tener nuevas tareas y proyectos* (7), el *salario* (5), la *conciliación de vida laboral /personal* (4), la *satisfacción laboral* (2), la *dificultad de la tarea* (3), la *carrera profesional* (3), la *formación continua* (3), la *motivación propia* (3), el *reconocimiento* (3) y otros elementos con menos de tres menciones. Con respecto a los desmotivadores, los participantes mencionaron la *rutina* (5), la *desigualdad entre el desempeño* (3), y la *desigualdad salarial para las mismas funciones* (3). Estos resultados se muestran gráficamente en la Ilustración 16.

Categoría	N	Ítems	N
Atracción	10	Proyectos y tareas nuevas	7
		Dificultad de la tarea	3
		Carrera profesional	3
		Viajes de trabajo	2
		Metodología	1
Satisfacción laboral	6	Satisfacción laboral	4
		Entorno de trabajo	2
		Estado de ánimo	1
		Instalaciones de trabajo	1
Dinero	6	Salario	5
		Participación en los beneficios de la organización	1
Formación	4	Formación continua	3
		Formación en nuevas tecnologías	1
Conciliación de la vida laboral/personal	4	Conciliación de la vida laboral/personal	2
		Horario reducido	2
Reconocimiento	3	Equipo	2
		Jefe	2
		Por el trabajo realizado	1
		Sentirse valorado	1
Autonomía	1	Autonomía	1
Otros	4	Auto motivación	3
		Buena relación con los miembros del equipo	1
		Motivación de los compañeros del equipo de trabajo	1
		Motivación del jefe	1

Tabla 18. Motivadores

Categoría	N	Ítems	N
Rutina	5	Rutina	5
Desigualdad	3	Desempeño	3
		Salario	3
		Compromiso	1
		Sin complementos salariales	1

Categoría	N	Ítems	N
Gestión	2	Bajo nivel de competencias de gestión en los jefes	1
		Estar parado	1
Desidia	1	Desidia	1

Tabla 19. Desmotivadores

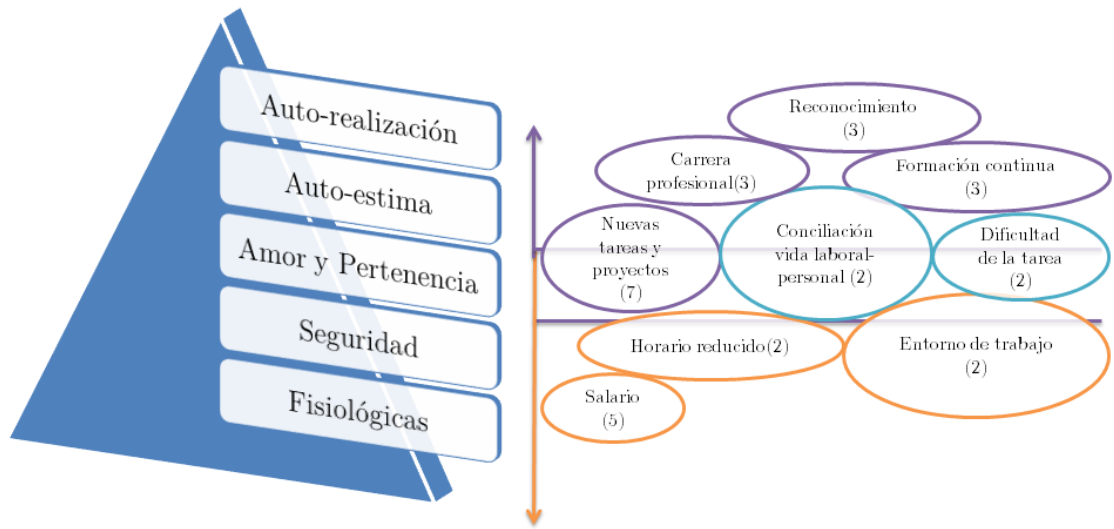


Ilustración 16. Mejora de la motivación

A modo de resumen, los elementos que mejoran la satisfacción laboral a partir de los resultados se ilustran gráficamente en Ilustración 17; en esta ilustración se muestran tres grupos principales de elementos: (1) organización del trabajo, (2) condiciones de trabajo, y (3) puesto de trabajo. Con respecto a la organización del trabajo, los elementos a tener en cuenta a la hora de mejorar la satisfacción laboral son el *equipo* (personas), la *relación de trabajo con los jefes y superiores*, la *comunicación con el resto de compañeros de trabajo* y el *cliente*, y la *formación recibida*. Además, el *entorno de trabajo* se enmarca dentro de la organización del trabajo aunque también podría considerarse como condiciones del trabajo ya que está relacionada con ambos conceptos. Con respecto a las condiciones de trabajo, el *salario*, las *condiciones contractuales*, y las *instalaciones de trabajo* son las principales fuentes de satisfacción laboral. Al igual que con el entorno de trabajo, el *horario* está vinculado también con la organización del trabajo por lo que se muestra vinculada con

ambos grupos. Finalmente, con respecto al propio puesto de trabajo, las *tareas a desempeñar* y el *cliente* son las principales fuentes. La tarea está relacionada también con la organización del trabajo y con las condiciones de trabajo, y el cliente está relacionado con la organización del trabajo.

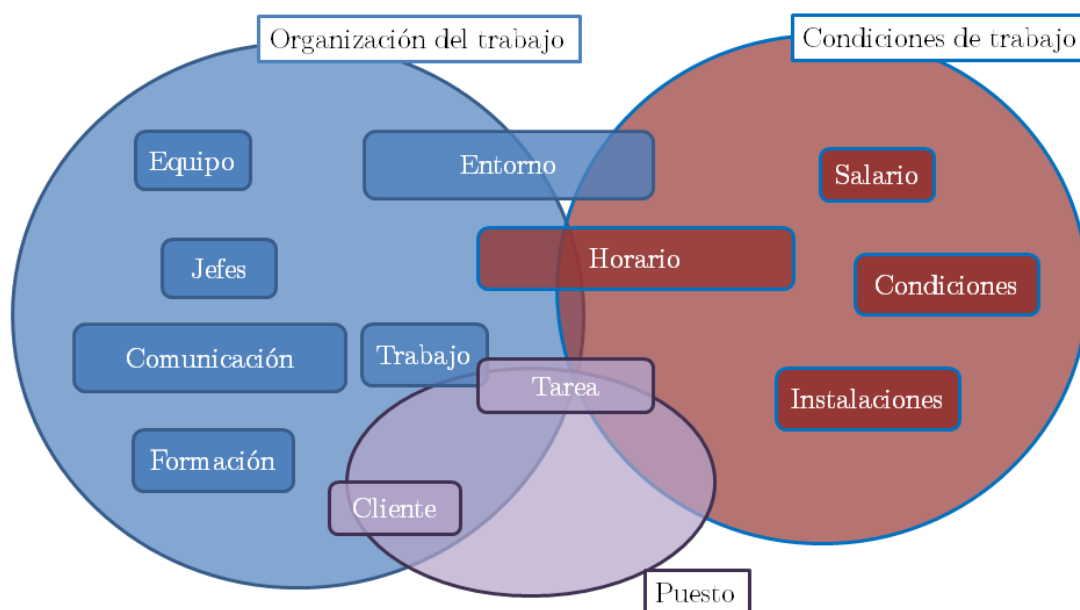


Ilustración 17. Principales elementos de la satisfacción laboral

3.3.2.3.1. *Discusión*

Los resultados obtenidos están en línea con las investigaciones previas en IS. Desde el punto de vista de la satisfacción laboral, los participantes indican una total satisfacción con sus compañeros de trabajo y miembros de equipo, pero no están totalmente satisfechos ni con sus jefes y superiores, ni con el tipo de trabajo que realizan. Además, la organización del trabajo es una fuente de desmotivación que debería ser analizada con más profundidad para conocer las causas de la baja satisfacción en este aspecto. Si los profesionales no están satisfechos con la gestión de sus responsables que ejercen sobre el trabajo (p. ej., la planificación, la gestión, y la distribución de tareas) empezarán a desmotivarse (Hall, Sharp, et al., 2008). Otro factor a tener en cuenta a partir de los resultados es el tipo de trabajo que se realiza

ya que la satisfacción con respecto a este elemento no es total y varios factores tales como la variedad de trabajo, la monotonía y la rutina fueron mencionados por los participantes. Estos problemas ya han sido mencionados en la literatura sobre motivación, y se deben considerar como modificadores de la misma en los profesionales de IS (Hall, Sharp, et al., 2008). Y no sólo en IS, la monotonía fue mencionada por Elton Mayo en relación con la organización industrial en los años 30, como un factor a tratar. También Herzberg señaló estos elementos al afirmar que "El trabajo debe ser suficientemente retador para utilizar todas las habilidades del empleado"¹¹. En esta dirección, el uso de metodologías y prácticas ágiles puede llevar a incrementar la satisfacción laboral ya que permite reducir algunos de estos elementos (Dyba & Dingsoyr, 2009) y tiene su justificación desde el punto de vista de los trabajadores del conocimiento (Horibe, 1999).

En relación con los elementos que mejoran la satisfacción laboral, hay una amplia mención al salario (un aumento del mismo) el cuál es uno de los principales motivadores extrínsecos para trabajar (Herzberg, et al., 1959) y también ha sido mencionado como un factor de insatisfacción laboral en IS en estudios anteriores (França & da Silva, 2010). Así pues, si entorno a la mitad de los participantes mencionan este elemento como una fuente de satisfacción, las organizaciones deberían considerar el uso del salario como una herramienta valiosa para incrementar la satisfacción laboral (Locke, Feren, McCaleb, Shaw, & Denny, 1980). Pero, hay que tener en cuenta que salarios más altos no producirá una mayor satisfacción laboral de forma directa (Judge, et al., 2010). Esta carencia se explica desde la pirámide de Maslow (1954) en la que el salario encaja en el segundo nivel, denominado seguridad. Una vez que el profesional alcanza el nivel salarial necesario, buscará otros elementos de jerarquía igual o superior que satisfacer, por ejemplo el reconocimiento y la responsabilidad. Además, desde el punto de vista de la teoría de factor dual (Herzberg,

¹¹ Traducción de "The job should have sufficient challenge to utilize the full ability of the employee."

et al., 1959), el salario se incluye dentro de los factores denominados higiénicos por lo que su ausencia no motiva, pero su presencia desmotiva; es decir, si un trabajador menciona el salario quiere decir que está insatisfecho con su salario, pero en caso de no mencionarlo no significa que le esté generando satisfacción. Hay que tener en cuenta que la satisfacción de los trabajadores del conocimiento se extiende más allá del salario (Horibe, 1999) al contrario de lo que Taylor pensaba sobre los trabajadores de comienzos de siglo XX, por lo que no se puede hablar sólo de salario cuando se gestiona la motivación y satisfacción laboral de los profesionales en IS. Tal y como Agarwal y Ferratt (2001) afirman "incluso las organizaciones que ofrecen salarios competitivos y trabajan con tecnologías punteras experimentan altos niveles de insatisfacción y mayores tasas de rotación (de personal) de las deseadas"¹². No obstante, factores intrínsecos, los cuales están en niveles más altos de la pirámide de Maslow y que predicen con mejor efectividad la satisfacción laboral que el salario (Judge & Church, 2000), también han sido mencionados. Por ejemplo, tener un trabajo menos monótono, la carrera profesional, el reconocimiento y la responsabilidad fueron mencionados por los participantes como fuentes de mejora de su satisfacción laboral. Estos factores han sido relacionados con la motivación en anteriores estudios en el área de IS (Hall, Sharp, et al., 2008). Desde el punto de vista Herzberg, "Los empleados que demuestran un incremento del nivel de sus habilidades deberían tener un mayor grado de responsabilidad"¹³, lo que está ampliamente relacionado con algunos de los elementos mencionados tales como la carrera profesional o la monotonía. Es posible que algunos de los participantes que mencionaron la responsabilidad cumplan los requisitos para desempeñar un puesto con mayor responsabilidad o para incluir tareas que requieran responsabilidad en su trabajo. Esto

¹² Traducción de *"even organizations that offer competitive salaries and work with leading-edge technologies experience high levels of dissatisfaction and higher than desired turnover"*

¹³ Traducción de *"Employees who demonstrate increasing levels of ability should be given increasing levels of responsibility."*

es importante dado que si los profesionales encuentran el trabajo como monótono y empiezan a sentir que pueden hacer más, empezaran a desmotivarse. Además, las organizaciones deberían considerar estos y otros motivadores intrínsecos, una vez que los extrínsecos están cubiertos, para mejorar la satisfacción laboral y reducir la tasa de rotación (Hulin, 1991) a la vez que incrementar los resultados económicos (Hayes, 1999).

Finalmente, es importante destacar que en los resultados está presente la relación de los conceptos analizados (satisfacción laboral, motivación y productividad) dado que varios de los conceptos fueron mencionados como motivadores y como fuentes de satisfacción laboral, y también de la productividad. Por ejemplo, el salario y la formación han sido mencionados en las tres categorías. Además, la satisfacción laboral fue mencionada como elemento motivador y algunos motivadores fueron mencionados como fuentes de satisfacción laboral, por ejemplo el reconocimiento o el contrato laboral. Hay que señalar que los resultados obtenidos son exploratorios. Dado que el objetivo de la tesis no se centra en la satisfacción laboral ni en la motivación, sino que se ha realizado el estudio de éstos como complemento, se da por finalizado el tratamiento de estos elementos para centrarse en los elementos que componen la medida de productividad que más tarde se elaborará y validará, es decir, en las entradas y salidas.

3.3.2.4. Factores que afectan a la productividad

Con respecto a la mejora de la productividad, los participantes mencionaron las *interrupciones* (3), el tener más *formación* (3), el *empleo de metodologías* (2), tener una *visión global del proyecto* (2), y otros elementos mencionados sólo una vez, tal y como se muestra en la Tabla 20. Estos elementos no son nuevos en el área de la IS y muchos de ellos ya han sido mencionados en la literatura tal y como se indica en la columna izquierda de dicha tabla.

Ítem	N	Investigaciones que también vinculan el elemento con la productividad
Menos interrupciones	3	(DeMarco & Lister, 1987; Jones, 2000)
Más formación	3	(Grady & Caswell, 1987; Turcotte & Rennison, 2004)
Metodología	2	(Boehm, 1981b; Dyba & Dingsoyr, 2009; Martin, 1981)
Visión global del estado del proyecto	2	-
Mejor planificación y gestión del proyecto	1	(Cole, 1995; Chatzoglou & Macaulay, 1997; Jones, 2000; Lakhanpal, 1993; Walston & Felix, 1977; Wohlin & Ahlgren, 1995)
Mejores instalaciones de trabajo (con áreas de descanso)	1	(DeMarco & Lister, 1987; Jones, 2000)
Más motivación	1	(Beecham, et al., 2008)
Desempeñar sólo un rol: analista, diseñador, programador	1	(DeMarco & Lister, 1987)
Mejor equipo de trabajo (PC)	1	(Vosburgh et al., 1984)
Mejor especificación de requisitos	1	(Berntsson-Svensson & Aurum, 2006; Finnie, Wittig, & Petkov, 1993; Maxwell & Forselius, 2000; Vosburgh, et al., 1984; Walston & Felix, 1977; Wohlin & Ahlgren, 1995)
Supervisar el impacto de los cambios solicitados por el cliente	1	(Berntsson-Svensson & Aurum, 2006; Finnie, et al., 1993; Maxwell & Forselius, 2000; Vosburgh, et al., 1984; Walston & Felix, 1977; Wohlin & Ahlgren, 1995)
Mejorar la comunicación con el cliente y con otros equipos de desarrollo	1	(Chatzoglou & Macaulay, 1997; Simmons, 1991; Wohlin & Ahlgren, 1995)
Incentivos	1	(McLean, Smits, & Tanner, 1996)
Más puestos de gestión	1	(Cole, 1995; Chatzoglou & Macaulay, 1997; Jones, 2000; Lakhanpal, 1993; Walston & Felix, 1977; Wohlin & Ahlgren, 1995)
Horario reducido	1	(Hill, Miller, Weiner, & Colihan, 1998; Jones, 2000)

Ítem	N	Investigaciones que también vinculan el elemento con la productividad
Satisfacción con el proyecto, equipo de trabajo y cliente	1	(Guzzo, 1988; Linberg, 1999; Rasch, 1991; Scudder & Kucic, 1991)
Anticipación	1	-
Flexibilidad de horarios	1	(Berntsson-Svensson & Aurum, 2006; Boehm, Abts, Brown, et al., 2000; Cole, 1995; DeMarco & Lister, 1987; Wohlin & Ahlgren, 1995)
Teletrabajo	1	(Hill, et al., 1998; Jones, 2000)

Tabla 20. Elementos que mejoran la productividad

Los resultados indican como elemento más mencionado a un factor que ya fue señalado hace años por dos de los pioneros en trabajar sobre el capital humano en IS DeMarco y Lister (1987), se trata de las interrupciones. Este factor puede medirse por ejemplo con el ratio que DeMarco y Lister llaman *E-Factor* que se calcula entre las horas interrumpidas y el tiempo presente en el puesto de trabajo, cuyos valores típicos oscilan entre 0.10 y 0,38 (DeMarco & Lister, 1987). Este elemento debe ser tenido en cuenta de forma especial debido a que las nuevas prácticas tales como el desarrollo ágil (Melnik & Maurer, 2006; Melo, et al., 2011; Tessem & Maurer, 2007), permiten la presencia del cliente en el proceso de desarrollo, lo que puede añadir interrupciones adicionales no planificadas. Además, tal y como uno de los participantes menciona, la influencia del cliente en el desarrollo es un riesgo alto cuyo impacto no puede ser fácilmente medido. Otras nuevas prácticas tales como el GSD pueden incrementar las interrupciones debido a diferencias de zona horaria y a problemas de comunicación (Guzmán, Ramos, Seco, & Esteban, 2011; Herbsleb & Moitra, 2001). Otros elementos mencionados son el disponer de *mejores instalaciones de trabajo* (con áreas de descanso), y tener una *mejor planificación y gestión de proyecto*, las cuales están bien recogidas en la literatura previa (Wagner & Ruhe, 2008b). Con respecto a las instalaciones de trabajo, se ha relacionado estrechamente con la configuración del puesto de trabajo, de forma que evite los sitios pequeños, ruidosos, y cargados de

personal, para aumentar el desempeño de los ingenieros software (DeMarco & Lister, 1987). Estos elementos también están mencionados como fuentes de satisfacción laboral y pueden ser considerados también como motivadores (Hall, Sharp, et al., 2008). Así pues, la relación entre motivación y productividad está presente en los resultados de esta fase de investigación, tal y como ha sido analizado en la literatura (Boehm, 1981b; Shepperd, 1993).

3.3.2.5. Conocimiento de las medidas convencionales por parte de los trabajadores

Dado que el estudio del estado de la cuestión indica que existían dos medidas ampliamente utilizadas para medir la productividad, y que una de ellas es mencionada como medida de productividad de los trabajadores de conocimiento bajo estudio (Ramirez & Nembhard, 2004), se preguntó a los participantes por el conocimiento y aplicabilidad de las medidas SLOC y PF por una unidad de tiempo o esfuerzo a su puesto de trabajo como medida de productividad. Los resultados fueron los siguientes:

- El 73.33% (11 participantes) conoce estas medidas, mientras que el otro 26.64% (4) las desconoce.
- De entre los participantes que las conocen:
 - Casi la mitad señalan que no sirve ninguna de las dos para medir la productividad de su puesto de trabajo (5/11, 45.45%).
 - Algo más de la mitad señalan que la medida basada en SLOC no sirve (6/11, 54.54%).
 - Y estos mismos participantes indican que la medida basada en PF si podría utilizarse (6/11, 54.54%), aunque habría que hacer modificaciones para que fuera eficaz.

Estos resultados tienen dos lecturas. Por un lado, las medidas de productividad basada en SLOC no tienen cabida en los puestos de los participantes y quizá tampoco

en muchos otros puestos de trabajo. Como se mencionó anteriormente, este tipo de medida sólo tiene en cuenta una de las salidas producidas y en la actualidad no tiene tanta importancia el número de líneas desarrolladas sino otra serie de elementos producidos y reutilizados. Por otro lado, las medidas basadas en PF pueden ser de utilidad tal y como mencionan Ramirez y Nembhard (2004) para medir la productividad de estos trabajadores del conocimiento. No obstante, no todos los participantes señalan esta medida como eficaz, y varios mencionan que debería ser ajustada, de modo que no es una medida universal para estos puestos de trabajo y es necesario un ajuste.

3.3.3. Contraste de hipótesis

A pesar de la naturaleza cualitativa de esta fase, es posible contrastar las hipótesis H1 y H3. La hipótesis H1 mantiene que *"En los puestos de trabajo involucrados en la ejecución de proyectos de desarrollo de software se emplean otras entradas, además del tiempo y el esfuerzo."* Esta hipótesis puede considerarse confirmada por los resultados obtenidos. Hay menciones tanto al tiempo como a otras entradas por parte de los participantes (p. ej., requisitos, conocimiento, cliente, recursos, documentación, experiencia...) lo que significa que otras entradas junto con el tiempo son utilizadas por los ingenieros software y jefes de proyecto para desempeñar las tareas de su puesto de trabajo. Por otro lado, la hipótesis H3 mantiene que *"En los puestos de trabajo involucrados en la ejecución de proyectos de desarrollo de software se producen otras salidas, además de líneas de código y la funcionalidad."* Esta hipótesis también puede considerarse confirmada por los resultados obtenidos. Hay menciones tanto al código fuente y funcionalidad como a otras salidas (p. ej., calidad, gestión del trabajo, documentación, ventas...) lo que indica que otras salidas junto con el código fuente y la funcionalidad son producidas. De este modo, H1 y H3 pueden considerarse confirmadas a pesar de no tener un resultado estadístico que apoye este contraste.

Con respecto a H2 y H4, no es posible contrastarlas en esta fase de la investigación, dada la reducida muestra utilizada, pero es posible arrojar algo de luz sobre el posible resultado. Por ejemplo, las ventas fueron mencionadas como salida por cuatro ingenieros software (36,36%) y por tres jefes de proyecto (75%), por lo que hace posible pensar que existe una diferencia en el grado de producción de esta salida (H4). Por otro lado, el código fuente previo fue mencionado por dos ingenieros software (18,18%) y no fue mencionado por ningún jefe de proyecto, de modo que H2 podría considerarse también confirmada. Para contrastar estas y las anteriores hipótesis, dado que los resultados hasta este momento parten de información cualitativa, se lleva a cabo el estudio cuantitativo descrito en la siguiente sección de la presente tesis (ver 3.4 Fase cuantitativa).

Con respecto a H5, los participantes que conocen las medidas de productividad basadas en SLOC y PF indican que las medidas basadas en SLOC no son válidas para medir la productividad de su puesto de trabajo y que las medidas basadas en PF podrían utilizarse realizándose un ajuste de las mismas al trabajo desempeñado. Así pues, esta hipótesis se considera apoyada por los resultados dado que estas medias son mejorables en el mejor de los casos, y las basadas en SLOC parecen no ser eficaces para medir la productividad de estos trabajadores.

3.3.4. Validez y fiabilidad

Debido a la naturaleza de la metodología empleada (estudio cualitativo con recogida de información mediante entrevistas) existen algunos sesgos que a continuación se mencionan. En primer lugar, la forma en la que los participantes fueron seleccionados produce un sesgo ya que los participantes se autoseleccionaron al recibir el email que les invitaba a participar. Este sesgo compromete la validez externa de los resultados obtenidos. En segundo lugar, el método de recogida de información (entrevista) produce una recogida de información subjetiva ya que está basada en una interacción entre el participante y el entrevistador que puede afectar a los datos de la

entrevista. Además de este sesgo es posible que se hayan producido sesgos de los siguientes tipos: (1) *deseabilidad social* debido a que los participantes han podido construir sus respuestas para ajustarse a las normas de su entorno y profesión, (2) *auto-representación* debido a que los participantes han podido comentar sus experiencias de una forma más negativa o positiva de lo que realmente fue, y (3) *plausibilidad* dado que parte de los sucesos contados han podido ser omitidos u olvidados, o reconstruidos con argumentos diferentes. No obstante, debido al carácter cualitativo de esta fase de la investigación se asume que los efectos de estos sesgos que pueden producir sobre la validez de los resultados son mínimos.

3.3.5. Conclusiones

Mediante el empleo de una metodología cualitativa basada en entrevistas a trabajadores de IS como método de recogida de información, y análisis de contenido como método para analizar la información, ha sido posible tener una visión conjunta sobre los objetivos de la presente investigación. Aunque este tipo de metodología requiere un gran esfuerzo, en comparación con las metodologías cuantitativas, la información obtenida es mucho más nutritiva para la investigación ya que aporta datos diversos de una misma muestra. De este modo ha sido posible obtener información sobre: la definición de productividad en tres niveles (organización, proyecto, e individual), las entradas y salidas utilizadas, la diferencia entre puestos, la motivación y satisfacción laboral, la mejora de la productividad, y el conocimiento de medidas de productividad basadas en SLOC o PF. Además, ha sido posible contrastar a partir de la información cualitativa algunas de las hipótesis planteadas.

Respecto a las definiciones obtenidas, existe una gran diferencia entre ellas y las establecidas por normas tales como la IEEE 1045-1992 que establecen una relación entre una salida y una entrada exclusivamente. A nivel de puesto de trabajo, la definición obtenida es más abstracta que la que se tiene en cuenta para crear las medidas más utilizadas de productividad a nivel de trabajador. A diferencia de la

definición aportada por la norma IEEE 1045-1992 la definición obtenida se centra en las tareas realizadas y no tanto en el propio desarrollo software. De este modo, las medidas basadas en SLOC o PF parecen tener una validez limitada, tal y como ya ha sido indicado en estudios previos (Petersen, 2011), y deberían ser consideradas sólo en las tareas en las que se realizara desarrollo de software exclusivamente. Además, a medida que la granularidad aumenta, la definición cambia y se centra en elementos de nivel superior. Por ejemplo, a nivel de proyecto las tareas dejan de tener significado y se incluye el concepto de proyecto como salida, y a nivel de organización se incluye el producto en pro del proyecto. Es importante destacar que el tiempo figura en las definiciones de todos los niveles, y que el concepto de productividad, a partir de los resultados, está vinculado con la estimación y planificación de proyectos. Otro elemento importante que figura en las definiciones, y que no está incluido en las medidas más utilizadas, es el concepto de calidad. Así pues, teniendo en cuenta estos resultados, parece necesario adaptar nuevas medidas de productividad a nivel de puesto de trabajo a la definiciones de productividad obtenidas.

Las entradas y salidas mencionadas por los participantes van más allá de la simplicidad con la que se diseñan las medidas de productividad basadas en SLOC y PF, dado que se han detectado otras entradas además del tiempo y el esfuerzo, y otras salidas además del código fuente y la funcionalidad. Esto indica que una medida de productividad debería considerar la diversidad de entradas y salidas potencialmente medibles de entre las existentes para dichos puestos y ajustarse a la realidad del mismo. Algunas de estas entradas y salidas pueden ser sustituidas por elementos proxy de modo que ciertos elementos podrían ser calculados a partir de otros. Además, la medida de productividad debería ser multi entrada y multi salida ya que la tipología y naturaleza de las entradas y salidas es diversa, y la utilización de una única entrada y salida como proxy para todas ellas es inviable. A su vez, los resultados sobre las salidas y entradas se relacionan con la definición de productividad a nivel de puesto, y varios elementos son tanto entradas como salidas. En dicha definición, la salida principal es la realización de la tarea y el cumplimiento de objetivos, junto con

la calidad, y la entrada principal es el tiempo. También, se mencionan diversos elementos relacionados con la gestión del trabajo, tanto como entradas como salidas.

Los participantes muestran una total satisfacción con sus compañeros de trabajo y miembros del equipo, pero no están del todo satisfechos ni con sus jefes, ni con el tipo de trabajo que realizan. Esto, junto a que la organización del trabajo es una fuente de desmotivación destacada por ellos que ha de ser tomada en cuenta, máxime cuando la productividad viene definida por la realización de las tareas del proyecto. En concreto, si los trabajadores no están satisfechos con la gestión realizada por sus superiores, empezarán a desmotivarse (Hall, Sharp, et al., 2008), y por lo tanto su productividad decaerá. Además, si los trabajadores no están satisfechos con el tipo de trabajo realizado, y, tal y como se ha obtenido en los resultados, señalan factores (higiénicos) como la variedad del trabajo, la monotonía y rutina, tratarlos para eliminarlos se convierte en una prioridad si se busca un aumento de la productividad. Estos resultados pueden interpretarse también como la necesidad de los trabajadores en IS de ser partícipes del proyecto, de modo que el empleo de metodologías ágiles puede ser muy útil en esta dirección (Dyba & Dingsoyr, 2009).

Desde el punto de vista de la mejora de la satisfacción, el aumento del sueldo es el elemento más mencionado. Este elemento es un motivador extrínseco además de factor higiénico, y ya ha sido mencionado en anteriores estudios dentro de IS (França & da Silva, 2010). Este elemento debería ser tratado con cuidado, ya que un aumento del mismo no producirá un aumento directo en la satisfacción, y por ende en la productividad (Judge, et al., 2010). No obstante, los participantes también mencionan factores motivadores tales como la carrera profesional, el reconocimiento, y la responsabilidad. Además, la relación entre los conceptos de productividad, satisfacción laboral y motivación está presente en los resultados dado que varios conceptos fueron mencionados en todas las categorías (p. ej., el salario y la formación). Así pues, las organizaciones han de considerar el estudio de los factores motivadores y de

satisfacción laboral como una fuente de conocimiento y mejora de la productividad de estos trabajadores.

Los resultados sobre mejora de la productividad están en línea con las anteriores investigaciones. Así, los elementos más mencionados por los participantes fueron las interrupciones, el disponer de más formación, el empleo de metodologías, y el tener una visión global del proyecto. Todos estos elementos han sido analizados previamente en la literatura científica y no resultan novedosos. No obstante, es necesario tener en cuenta los factores que afectan a la productividad de los trabajadores del conocimiento a la hora de realizar una medición. En este sentido, y dado que cada puesto de trabajo se desempeña bajo condiciones distintas a otros, aunque sean del mismo tipo, cualquier estudio de productividad debería tener en cuenta estas condiciones a la hora de valorar e interpretar los resultados obtenidos.

Con respecto al conocimiento de las medidas de productividad basadas en SLOC o PF, casi una cuarta parte de los participantes no las conocen. De las otras tres cuartas partes, algo más de la mitad (un tercio de la muestra) señalan que las medidas basadas en SLOC carecen de eficacia para medir la productividad de su trabajo, pero que las basadas en PF podrían aplicarse con ciertas modificaciones; la otra mitad indica que ninguna de estas medidas tiene eficacia para medir la productividad de su puesto. Estos resultados indican que las medidas de productividad basadas en SLOC deberían ser descartadas para medir la productividad en los puestos de trabajo analizados, y que las basadas en PF deberían ser modificadas para ajustarse a la realidad de las tareas desempeñadas por los trabajadores.

Finalmente, no ha sido posible contrastar las hipótesis estadísticamente, pero si ha sido posible contrastar las hipótesis H1 y H3 desde la información cualitativa obtenida. A partir de los resultados, es razonable suponer que existen otras entradas además del tiempo y el esfuerzo (H1), y otras salidas además del código fuente y la funcionalidad (H3), ya que se mencionan otras entradas y salidas. De este modo, H1 y H3 pueden considerarse confirmadas, a pesar de no tener un resultado estadístico que

apoye este contraste. Así pues, es necesario continuar con la investigación para contrastar las hipótesis H1 a H4 de forma estadística, y posteriormente desarrollar medidas de productividad basándose en los resultados obtenidos. En la Tabla 21 se muestra el estado de las hipótesis tras realizar la fase cualitativa.

Hipótesis	Contrastada	Resultado
Hipótesis 1. En los puestos de trabajo involucrados en la ejecución de proyectos de desarrollo de software se emplean otras entradas, además del tiempo y el esfuerzo.	Sí, desde un punto de vista exploratorio. No obstante, será contrastada de nuevo a raíz de los resultados del cuantitativo con rigor estadístico.	Sí, previamente a la realización de la fase cuantitativa tiene sentido suponer que existen otras entradas (ver Tabla 13).
Hipótesis 2. Las entradas utilizadas son distintas para cada puesto de trabajo involucrado en la ejecución de proyectos de desarrollo de software.	No, ya que no fue posible contrastarla mediante la fase cualitativa. Se contrastará en la fase cuantitativa.	Pendiente.
Hipótesis 3. En los puestos de trabajo involucrados en la ejecución de proyectos de desarrollo de software se producen otras salidas, además de líneas de código y la funcionalidad.	Sí, desde un punto de vista exploratorio. No obstante, será contrastada de nuevo a raíz de los resultados del cuantitativo con rigor estadístico.	Sí, previamente a la realización de la fase cuantitativa tiene sentido suponer que existen otras entradas (ver Tabla 14).
Hipótesis 4. Las salidas producidas son distintas para cada puesto de trabajo involucrado en la ejecución de proyectos de desarrollo de software.	No, ya que fue no posible contrastarla mediante la fase cualitativa. Se contrastará en la fase cuantitativa.	Pendiente.

Hipótesis	Contrastada	Resultado
Hipótesis 5. Las actuales medidas de productividad a nivel de puesto de trabajo en los proyectos de desarrollo de software tienen una eficacia limitada para medir la productividad real de los trabajadores.	Sí, desde un punto de vista exploratorio. Aunque parcialmente se ha contrastado esta hipótesis en la fase cualitativa, será contrastada tras el estudio empírico.	No, sin embargo, y previamente a la realización de la validación, tiene sentido suponer que las medidas basadas en SLOC o PF por una unidad de esfuerzo o tiempo no son eficaces para medir la productividad de todos los puestos.
Hipótesis 6. Es posible medir de forma más eficaz la productividad de los puestos de trabajo en los proyectos de desarrollo de software con nuevas medidas que combinen varios elementos: entradas, salidas y factores.	No, será contrastada tras el estudio empírico.	Pendiente.

Tabla 21. Estado de las hipótesis tras ejecutar la fase cualitativa

3.4. Fase cuantitativa

Los resultados obtenidos en la fase cualitativa permiten responder a algunas de las hipótesis planteadas y otras quedan parcialmente contratadas (ver Tabla 21). En concreto, se han contrastado las hipótesis relacionadas con la existencia de otras entradas, además del tiempo y el esfuerzo (H1), y la existencia de otras salidas, además del código fuente y la funcionalidad (H3). Además, dada la existencia de otras entradas y salidas, que no son medidas en las medidas actuales, tiene sentido suponer que la eficacia de las medidas más utilizadas para medir la productividad de estos trabajadores es limitada (H5). Así, quedan pendientes por contrastar las hipótesis de investigación H1, H2, H3 y H4. Para conseguir el objetivo de contrastar las hipótesis, se ha diseñado y ejecutado el método cuantitativo que se presentan a continuación, junto con los resultados obtenidos y el contraste de las hipótesis realizado.

3.4.1. Metodología

El método elegido está basado en el análisis estadístico para contrastar las hipótesis y el uso de un cuestionario como herramienta de recogida de información. En las ciencias sociales es común utilizar los cuestionarios, los cuales son un tipo de encuesta que solicita a los participantes definir el grado con el que están de acuerdo o desacuerdo sobre varios ítems (Foddy, 1994); estas escalas de acuerdo suelen usar intervalos (McIver & Carmines, 1981). En IS, los cuestionarios han sido también utilizados para la investigación (Ciolkowski, Laitenberger, Vegas, & Biffel, 2003). El proceso utilizado en esta fase se muestra en la Ilustración 18.

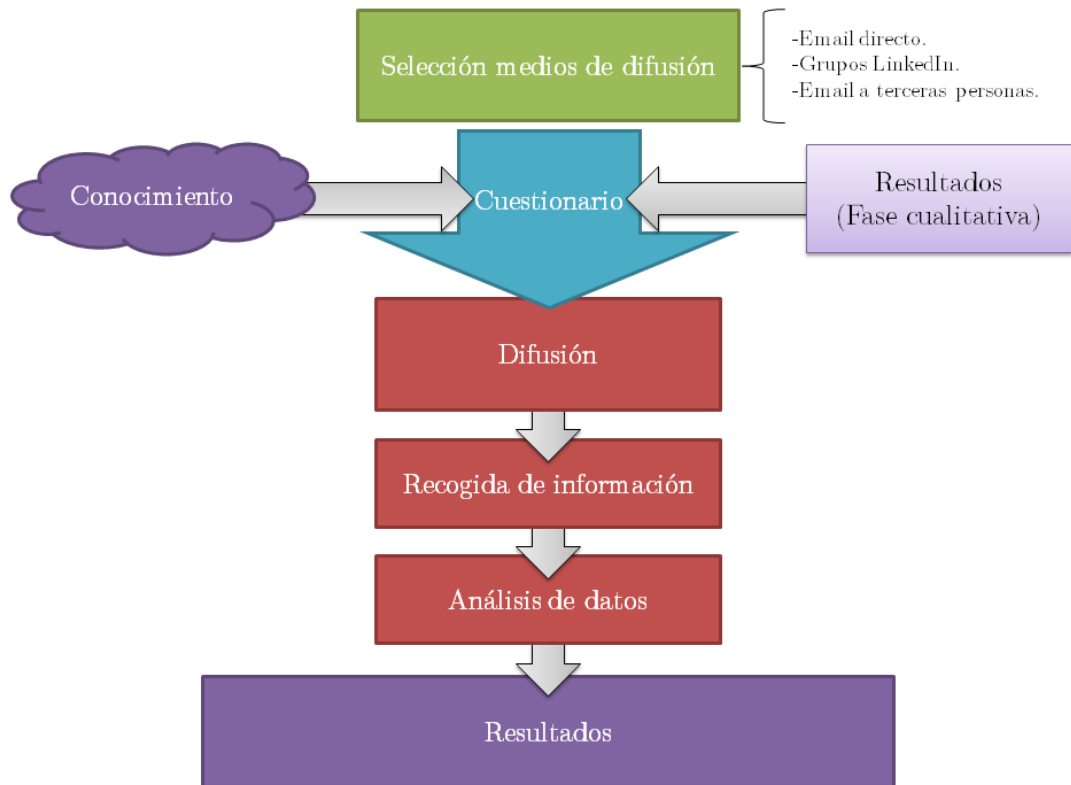


Ilustración 18. Proceso fase cuantitativa

A partir de los resultados de la fase cualitativa, se construyó un cuestionario como forma de recogida de información. El cuestionario incluyó preguntas demográficas para describir la muestra utilizada, y 32 preguntas con escala Likert (1932) de 6 valores ("Nunca", "Casi Nunca", "A veces", "A menudo", "Casi Siempre", "Siempre") para los

cuatro puestos de trabajo que Métrica 3 (2000) define en relación con los proyectos de desarrollo de software: programador, analista, consultor, y jefe de proyecto. Dentro de cada puesto, los participantes rellenaron 32 preguntas, 16 relacionadas con el grado de utilización de recursos (entradas) y 16 con el grado de producción de productos o servicios (salidas) para los trabajos en los que tuvieran experiencia. Los elementos seleccionados fueron los siguientes, como entradas: *Tiempo, Conocimiento, Planificación, Estimación, Objetivos asignados, Software, Hardware, Instalaciones de trabajo, Especificación de requisitos, Conocimiento funcional, Cliente, Motivación, Documentación, Experiencia, Formación, Código fuente previo*; y como salidas: *Producto, Documentación, Tarea finalizada, Cumplimiento de objetivos, Estimación, Planificación, Calidad, Ventas, Pruebas, Experiencia, Conocimiento, Solución de problemas, Solución de bugs, Satisfacción del cliente, Funcionalidad*. Estas listas no son limitadas, es decir, no son todas las posibles entradas y salidas, pero representan una muestra significativa de ellas, y permiten el contraste estadístico de las hipótesis de investigación. El formulario se puede consultar en el Anexo D - Cuestionario y justificación de selección de elementos, junto con la justificación de la selección de las entradas y salidas.

Para el tratamiento de datos se ha utilizado el paquete estadístico SPSS versión 17. Y como método estadístico para comprobar las hipótesis el test no-paramétrico de Kruskal-Wallis (Kruskal & Wallis, 1952) con el objetivo de conocer si existen diferencias entre en el grado de utilización de las entradas para cada puesto de trabajo y en el grado de producción de la salida. Este test fue elegido, en sustitución del test ANOVA de una vía, ya que aunque los datos a analizar son de tipo ordinal (escala Likert) y podrían considerarse valores numéricos, los datos no cumplen las otras dos condiciones necesarias para aplicar ANOVA: normalidad y homocedasticidad¹⁴. En el

¹⁴ Las varianzas de la variable dependiente en los grupos que se comparan son aproximadamente iguales.

Anexo E - Condiciones para aplicar ANOVA se incluyen las pruebas de normalidad y homocedasticidad que justifican la selección del estadístico de contraste. Además del test de Kruskal-Wallis se eligió el test post hoc de Dunn, el cuál es el análogo en formato no-paramétrico de la prueba Holm-Sidak de múltiple t-test, para comparar las diferencias entre los puestos de trabajo para las entradas, y para las salidas.

3.4.2. Recogida muestral

El cuestionario fue distribuido utilizando la plataforma *Google Docs*, que permite la construcción y rellenado de cuestionarios electrónicos online. El cuestionario fue realizado en español y traducido al inglés. Para enviar el formulario se utilizaron dos medios. Por un lado, se mando un email a contactos personales; y por otro lado, se puso un *post* en varios grupos de *LinkedIn*. En concreto, se envió el formulario a 300 profesionales y se posteo en 36 grupos. El cuestionario incluye una pregunta específica para saber cómo han accedido los participantes al mismo, de forma que es posible conocer el ratio de respuesta para cada medio utilizado. Como respuestas a esta pregunta, además de los dos medios utilizados, se incluye una tercera opción que permite a los participantes indicar que han recibido el email de una tercera persona. Esta opción fue incluida dado que se solicitaba a los posibles participantes que enviaran el email, o que facilitaran la dirección en el caso de los grupos LinkedIn, a personas conocidas que pudieran rellenar el formulario.

Dado que la recogida muestral se ha realizado mediante la herramienta de *Google Docs*, los datos se trasladaban automáticamente a una hoja de cálculo. Se fijó un número orientativo de participantes mínimo en 100 para comenzar a analizar los datos, no obstante se obtuvieron 158 participaciones. La muestra final tiene las siguientes características: los participantes rellenaron un total de 345 puestos (125 programador, 95 analista, 65 consultor, y 60 jefe de proyecto), es decir, 2,18 puestos por participante; con respecto a la composición demográfica de la muestra, ésta consta de 131 hombres (82,91%) y 27 mujeres (17,09%), con una media de 33,94 años con

desviación típica de 8,62 años; y el medio que utilizaron para acceder los participantes se compone de la siguiente manera: 89 participantes (56,33%) accedieron mediante email, lo que representa un ratio de respuesta del 24% ya que se enviaron 300 emails, 49 (31,01%) mediante los grupos de LinkedIn, y 20 (12,66%) mediante una persona conocida.

3.4.3. Resultados

A continuación se presentan los resultados en dos apartados. En primer lugar, los resultados en formato descriptivo para las cuestiones sobre entradas y salidas, y los datos demográficos; y en segundo lugar el contraste de hipótesis.

3.4.3.1. Descriptivos

Los resultados descriptivos están incluidos en la Tabla 22 para las entradas y la Tabla 23 para las salidas. Se incluye para cada ítem la moda y los percentiles: 25, 50 (mediana), y 75. Dado que se eligió una escala Likert, la cual debe ser considerada inicialmente como ordinal y no de intervalo, y no hay motivos para considerarla como intervalo (Kitchenham, Hughes, & Linkman, 2001), no se incluye la media ni la desviación típica al no ser datos continuos ni intervalo¹⁵. Así pues, se emplea la moda y percentiles como estadísticos descriptivos. Destacar, que algunos elementos tienen múltiples modas; en estos casos se muestra la moda mínima. Además, se incluyen la

¹⁵ Sólo en casos en los que los elementos de la escala Likert conforman datos de tipo intervalo es correcto calcular la media y desviación típica de los datos obtenidos. En este caso, la distancia entre elementos de la escala Likert empleada no puede ser comparada por lo que está fuera de lugar emplear estas medidas de centralización y dispersión al tratarse de una escala de tipo ordinal; para más información sobre esta peculiaridad consultar: (Andrews, Klem, Davidson, O'Malley, & Rodgers, 1981; Blaikie, 2003; Clegg, 1998).

desviación semi-intercuartílica y la desviación mediana para cada ítem, en la Tabla 24 para las entradas y en la Tabla 25 para las salidas.

A partir de los datos obtenidos se puede decir que todas las entradas son ampliamente utilizadas por todos los puestos bajo estudio. En concreto, todos los elementos tienen una mediana igual o superior a 4 ("A menudo") salvo *Código fuente previo* que tiene una mediana de 2 ("Casi Nunca") para el puesto Jefe de Proyecto y Consultores, y de 3 ("A veces") para Analistas y Programadores. Desde el punto de vista de las salidas casi todas ellas tienen una mediana igual o superior a 4 salvo: las *Ventas* para el puesto de Programador y Analista tienen una media de 3, el *Código fuente* para el Analista de 3 ("A veces") y para el Consultor y Jefe de Proyecto de 2 ("Casi Nunca"), los *Bugs solucionados* para el Analista, Programador y el Jefe de Proyecto de 3 ("A veces"), y el *Producto* para el Consultor de 3 ("A veces"). Así pues, la selección de entradas y salidas puede considerarse válida para los puestos de trabajo bajo estudio.

	Programador (n=125)				Analista (n=95)				Consultor (n=65)				Jefe de Proyecto (n=60)			
	Moda	Percentiles			Moda	Percentiles			Moda	Percentiles			Moda	Percentiles		
		25	50	75		25	50	75		25	50	75		25	50	75
Tiempo	6	4	5	6	6	4	5	6	6	5	6	6	6	5	6	6
Conocimiento	6	5	6	6	6	5	5	6	6	5	6	6	6	5	6	6
Planificación	4*	4	5	5	5	4	5	6	5	4	5	6	6	5	6	6
Estimación	4	3	4	5	5	4	5	6	5	4	5	6	6	5	5,5	6
Objetivos asignados	5	4	5	5	5	4		5	5	4	5	6	6	5	6	6
Software	6	5	6	6	5	4	5	5	5*	3	4	5,5	5*	3	5	6
Hardware	6	3	5	6	5	2	4	5	2	2	4	5	4	2	3	4
Instalaciones de trabajo	5	4	5	5,5	4	3	4	5	5	3	4		5	3	4	5
Especificación de requisitos	5	3	5	5	6	4	5	6	5	3	4	5	6	4	5	6
Conocimiento funcional	5	4	5	5	6	5	5	6	6	4	5	6	5	4	5	6
Cliente	4	3	4	5	6	4	5	6	6	4,5	5	6	5*	4	5	6
Motivación	5	4	5	5	4	4	4	5	5	4	5	6	6	4	5	6
Documentación	5	4	5	5	5	4	5	6	5	2,5	5	5	5	4	5	6
Experiencia	5	4	5	6	6	4	5	6	6	5	5	6	6	4	5	6
Formación	5	4	5	5	5	4	4	5	5		5	6	4*	3	4	5
Código fuente previo	4	3	4	5	3	2	3	4	2	2	2	4	2	2	2	4

Tabla 22. Estadísticos descriptivos de las entradas

a. Existen múltiples modas por lo que se muestra el valor menor

	Programador (n=125)				Analista (n=95)				Consultor (n=65)				Jefe de Proyecto (n=60)			
	Moda	Percentiles			Moda	Percentiles			Moda	Percentiles			Moda	Percentiles		
		25	50	75		25	50	75		25	50	75		25	50	75
Código fuente	6	5	5	6	2	2	3	4	1	1	2	4	1	1	2	4,75
Producto	5	4	5	5	5	3	4	5	2*	2	3	5	5	2	4	5
Documentación	5	3,5	4	5	6	5	5	6	6	4	5	6	5	4	5	6
Tarea finalizada	5	4	5	5	5	4	5	6	5	4	5	6	6	5	5	6
Cumplimiento de objetivos	5	4	5	5	5	4	5	6	5	4	5	6	5	5	5	6
Estimación	4	3	4	5	5	4	5	5	4*	4	5	5	5	4,25	5	6
Planificación	4*	3	4	5	5	4	5	6	5	4	5	5	6	5	5	6
Calidad	5	4	5	5	5	4	5	6	5*	4	5	6	5	4	5	6
Ventas	2	1	2	3	2	2	3	4	5	2	4	5	4	2,25	4	4
Pruebas	5	4	5	5	5	3	4	5	4	2	4	5	5	3	4	5
Experiencia	5	4	5	5,5	5	4	5	6	6	4	5	6	6	4,25	5	6
Conocimiento	5	4	5	6	5	4	5	6	6	5	5	6	6	4	5	6
Solución de problemas	5	4	5	6	5	4	5	6	5	4	5	6	5	4	5	6
Solución de bugs	5	4	5	5	3	2	3	4	2	2	3	4	3	2	3	4
Satisfacción del cliente	5	3	4	5	5	4	5	5	5	4	5	6	5	5	5	6
Funcionalidad	5	4	5	6	5	4	5	6	5*	4	5	6	6	4	5	6

Tabla 23. Estadísticos descriptivos de las salidas

a. Existen múltiples modas por lo que se muestra el valor menor

Ítem	Desviación Semi-Intercuartílica (Q)				Desviación Mediana (DM)			
	Programador	Analista	Consultor	Jefe de Proyecto	Programador	Analista	Consultor	Jefe de Proyecto
Tiempo	0,5	1	0,5	0,5	0,824	0,874	0,577	0,500
Conocimiento	0,5	0,5	0,5	0,5	0,580	0,737	0,492	0,533
Planificación	1	1	1	0,5	0,880	0,811	1,031	0,983
Estimación	1	1	1	0,5	0,960	0,811	1,015	0,750
Objetivos asignados	1	0,5	1	0,5	0,944	0,642	0,815	0,583
Software	1	0,5	1,25	1,5	0,580	0,874	1,092	1,200
Hardware	1,75	1,5	1,5	1	1,344	1,326	1,385	1,300
Instalaciones de trabajo	1	1	1	1	1,000	1,042	1,092	1,200
Especificación de requisitos								
Conocimiento funcional	1	0,5	1	1	1,000	0,811	0,985	0,900
Cliente	1	1	0,75	1	0,800	0,758	0,800	0,700
Motivación	1	0,5	1	1	1,008	1,032	0,831	0,857
Documentación	0,5	1	0,75	1	1,056	1,042	0,831	0,883
Experiencia	0,5	1	0,5	1	0,920	0,853	0,985	0,917
Formación	0,5	0,5	1	1	0,728	0,705	0,708	0,850
Código fuente previo	1	1	1	1	1,016	1,021	0,969	1,133
					0,872	0,863	1,138	1,117

Tabla 24. Desviación semi-intercuartílica y mediana de las entradas

Ítem	Desviación Semi-Intercuartílica (Q)				Desviación Mediana (DM)			
	Programador	Analista	Consultor	Jefe de Proyecto	Programador	Analista	Consultor	Jefe de Proyecto
Código fuente	1,5	1	1,5	1,875	0,864	1,211	1,431	1,433
Producto	1	1	1,5	1,5	0,936	1,263	1,354	1,383
Documentación	1	0,5	1	1	0,992	0,832	0,985	0,833
Tarea finalizada	1	1	1	0,5	0,616	0,926	0,877	0,800
Cumplimiento de objetivos	1	1	1	0,5	0,728	0,821	0,815	0,617
Estimación	0,5	0,5	0,5	0,875	0,920	0,832	0,969	0,717
Planificación	0,5	1	0,5	0,5	0,880	0,716	0,877	0,733
Calidad	1	1	1	1	0,968	0,800	0,923	0,717
Ventas	1	1	1,5	0,875	0,952	1,221	1,262	1,100
Pruebas	1	1	1,5	1	0,856	1,116	1,246	1,133
Experiencia	1	1	1	0,875	0,736	0,705	0,754	0,800
Conocimiento	1	1	0,5	1	0,840	0,705	0,754	0,850
Solución de problemas	1	1	1	1	0,672	1,105	0,892	0,933
Solución de bugs	1	1	1	1	0,944	1,137	1,215	1,200
Satisfacción del cliente	0,5	0,5	1	0,5	1,008	0,947	0,723	0,700
Funcionalidad	1	1	1	1	0,696	0,811	1,062	0,967

Tabla 25. Desviación semi-intercuartílica y mediana de las salidas

3.4.4. Contraste de hipótesis

Respecto a las Hipótesis H1 y H3, los resultados constatan la existencia de otras entradas, además del esfuerzo y el tiempo, y otras salidas, además del código fuente y la funcionalidad. Así, las hipótesis

Hipótesis 1. En los puestos de trabajo involucrados en la ejecución de proyectos de desarrollo de software se emplean otras entradas, además del tiempo y el esfuerzo.

Hipótesis 3. En los puestos de trabajo involucrados en la ejecución de proyectos de desarrollo de software se producen otras salidas, además de líneas de código y la funcionalidad.

quedan confirmadas. Esto se basa en la utilización de varias de las entradas planteadas con una frecuencia alta por parte de todos los puestos, y la producción de varias de las salidas planteadas con una frecuencia alta por parte de todos los puestos. Dado que son hipótesis de existencia no es posible aplicar ningún método estadístico de contraste y sólo es posible basarse en los resultados descriptivos descritos en el apartado anterior.

Para contrastar las hipótesis H2 y H4, referentes a las diferencias entre puestos de trabajo en cuanto al empleo de las entradas (H2) y a la producción de las salidas (H4), es necesario emplear un método estadístico, en este caso se optó por emplear el test no-paramétrico de Kruskal-Wallis (Kruskal & Wallis, 1952). Además del test de Kruskal-Wallis se eligió el test post hoc de Dunn, el cuál es el análogo en formato no-paramétrico de la prueba Holm-Sidak de múltiple t-test, para comparar las diferencias entre los puestos de trabajo para las entradas, y para las salidas. Los resultados del test de Kruskal Wallis están incluidos en la Tabla 26.

Desde el punto de vista de las entradas: el *Tiempo* ($\chi^2 = 14,752$, $df = 3$, $P < ,005$), ($\chi^2 = 14,752$, $df = 3$, $P < ,005$), la *Planificación* ($\chi^2 = 29,05$, $df = 3$, $P < ,005$), la *Estimación* ($\chi^2 = 34,986$, $df = 3$, $P < ,005$), los *Objetivos asignados* ($\chi^2 = 24,825$, $df = 3$, $P < ,005$), el *Software* ($\chi^2 = 44,532$, $df = 3$, $P < ,005$), el *Hardware* ($\chi^2 = 28,591$, $df = 3$, $P < ,005$), la *Especificación de requisitos* ($\chi^2 = 19,427$, $df = 3$, $P < ,005$), el *Conocimiento funcional* ($\chi^2 = 11,019$, $df = 3$, $P < ,005$), el *Cliente* ($\chi^2 = 41,015$, $df = 3$, $P < ,005$), la *Motivación* ($\chi^2 = 9,121$, $df = 3$, $P < ,005$), y el *Código fuente previo* ($\chi^2 = 62,451$, $df = 3$, $P < ,005$) tienen diferencias estadísticas significativas entre algunas parejas de puesto de trabajo, o lo que es lo mismo, alguno puestos de trabajo emplean estos recursos con distinto grado de frecuencia.

Por otro lado, desde el punto de vista de las salidas: el *Código fuente* ($\chi^2 = 106,906$, $df = 3$, $P < ,005$), el *Producto* ($\chi^2 = 22,771$, $df = 3$, $P < ,005$), la *Documentación* ($\chi^2 = 25,846$, $df = 3$, $P < ,005$), los *Objetivos Alcanzados* ($\chi^2 = 8,617$, $df = 3$, $P < ,005$), la *Estimación* ($\chi^2 = 37,112$, $df = 3$, $P < ,005$), la *Planificación* ($\chi^2 = 55,848$, $df = 3$, $P < ,005$), la *Calidad* ($\chi^2 = 4,587$, $df = 3$, $P < ,005$), *Sales* ($\chi^2 = 35,913$, $df = 3$, $P < ,005$), las *Pruebas* ($\chi^2 = 23,435$, $df = 3$, $P < ,005$), la *Solución de Bugs* ($\chi^2 = 58,192$, $df = 3$, $P < ,005$), y *Satisfacción del cliente* ($\chi^2 = 28,105$, $df = 3$, $P < ,005$) tienen diferencias estadísticas significativas, por lo que para estas salidas es posible decir que son producidas con distinto grado de frecuencia en los puestos de trabajo.

		Ji Cuadrado	Sig. asint.
Entradas	Tiempo	14.752	,002
	Conocimiento	6.29	,098
	Planificación	29.05	,000
	Estimación	34.986	,000
	Objetivos asignados	24.825	,000
	Software	44.532	,000
	Hardware	28.951	,000
	Instalaciones de trabajo	7.075	,070
	Especificación de requisitos	19.427	,000

		Ji Cuadrado	Sig. asint.
	Conocimiento funcional	11.019	,012
	Cliente	41.051	,000
	Motivación	9.121	,028
	Documentación	4.8	,187
	Experiencia	3.906	,272
	Formación	4.43	,219
	Código fuente previo	62.451	,000
		Ji Cuadrado	Sig. asint.
Salidas	Código fuente	106.906	,000
	Producto	22.771	,000
	Documentación	25.846	,000
	Tarea finalizada	5.968	,113
	Cumplimiento de objetivos	8.617	,035
	Estimación	37.112	,000
	Planificación	55.848	,000
	Calidad	4.587	,205
	Ventas	35.913	,000
	Pruebas	23.435	,000
	Experiencia	6.256	,100
	Conocimiento	5.2	,158
	Solución de problemas	5.572	,134
	Solución de bugs	58.192	,000
	Satisfacción del cliente	28.105	,000
	Funcionalidad	3.522	,318

Tabla 26. Resultados del test de Kruskal Wallis (variable de agrupación: puesto de trabajo)

Muchas de las entradas utilizadas en el cuestionario (11/16) tienen diferencias estadísticamente significativas entre algunos de los puestos de trabajo por lo que la hipótesis H3 *"Las entradas utilizadas son distintas para cada puesto de trabajo involucrado en la ejecución de proyectos de desarrollo de software."* puede considerarse confirmada a partir de estos resultados. Por otro lado, varias salidas (10/16) tienen diferencias estadísticamente significativas entre algunos de los puestos

de trabajo por lo que la hipótesis H4 "*Las salidas producidas son distintas para cada puesto de trabajo involucrado en la ejecución de proyectos de desarrollo de software.*" puede considerarse confirmada a partir de estos resultados.

Adicionalmente, para conocer entre que pares de puestos de trabajo existen las diferencias en la utilización de las entradas y en la producción de las salidas, se llevó a cabo el test de Dunn. El software estadístico utilizado (*SPSS versión 17*) no incluye este test por lo que fue construido utilizando una hoja de cálculo de *Microsoft Excel*.

El primer paso para construir el test de Dunn es establecer el nivel de significancia para el mismo (Ecuación 1) cuyo valor es 0,004166667.

$$\alpha = \frac{\alpha'}{K(K-1)}$$

Ecuación 1. Nivel de significancia del test de Dunn

donde

α' es el alpha ajustado utilizado en la prueba de Kruskal Wallis (0,05),

y K es el número de grupos (4)

El siguiente paso es establecer la diferencia teórica para cada par de grupos a comparar (Ecuación 2). Los resultados de este paso se muestran en la Tabla 27.

$$\Delta_{ij} = Z_{1-\alpha} \sqrt{\frac{N(N+1)}{12} \left(\frac{1}{n_i} + \frac{1}{n_j} \right)}$$

Ecuación 2. Diferencias teóricas para cada pareja de grupos a comparar

donde

$Z_{1-\alpha}$ es la inversa de la distribución normal acumulada (2,638257273),

N es el tamaño de la muestra ($N=345$),

y n_i y n_j son los tamaños de los grupos que son comparados (125, 95, 65, 60).

	Analista	Consultor	Jefe de Proyecto
Programador	35.81530462	40.2382638	41.326569
Analista		42.3560672	43.3912878
Consultor			52.3508412

Tabla 27. Diferencias teóricas (prueba de Dunn)

El último paso consiste en calcular la diferencia observada para cada elemento en cada grupo (Ecuación 3). Si la diferencia observada es mayor que la diferencia teórica (Ecuación 2), entonces hay una diferencia significativa entre los grupos para ese elemento. Las diferencias observadas junto con los rangos medios para cada elemento y par de puestos de trabajo están incluidas en la Tabla 28 para las entradas, y en la Tabla 29 para las salidas.

$$\Delta'_{ij} = |\overline{R}_i - \overline{R}_j|$$

Ecuación 3. Diferencia observada para cada pareja de grupos comparados

donde

\overline{R}_i y \overline{R}_j son los rangos medios de los dos grupos a comparar.

En el caso de la motivación (entrada) no se encontró diferencia entre ningún par a pesar de que la prueba de Kruskal Wallis indicaba la existencia de al menos una par con diferencias. Así pues se decidió utilizar la prueba de U Mann-Whitney para esta entrada. Con esta prueba se encontraron diferencias significativas para los siguientes pares: Programador y Consultor ($z = -2.335$, $p < .05$), Programador y Jefe de Proyecto ($z = -2.129$, $p < .05$), y Analista y Consultor ($z = -2.079$, $p < .05$) para esta entrada.

Con respecto a las diferencias entre puestos y las entradas, el puesto de trabajo con más diferencias con respecto al uso de ellas es Programador, con 21 diferencias estadísticamente significativas (de 96 posibles); seguido del puesto Jefe de Proyecto con 16 diferencias. Además, se puede observar que para algunas entradas hay

diferencias entre varios puestos de trabajo. Por ejemplo, el grado de utilización de la *Planificación* y los *Objetivos asignados* por el Jefe de Proyecto es estadísticamente diferente con respecto a los otros puestos. El mismo patrón se reproduce para el programador con respecto al grado de uso del *Cliente*, *Software*, *Hardware*, y *Código fuente previo*.

Ítem	Puesto de Trabajo	Rango Medio	Diferencias observadas		
			Analista	Consultor	Jefe de Proyecto
Tiempo	Programador	159.67	0.76	28.40	47.06*
	Analista	158.92		29.16	47.82*
	Consultor	188.08			18.66
	Jefe de Proyecto	206.73			
Planificación	Programador	147.69	31.32	15.13	79.57*
	Analista	179.01		16.19	48.25*
	Consultor	162.82			64.44*
	Jefe de Proyecto	227.26			
Estimación	Programador	140.85	40.45*	29.89	88.41*
	Analista	181.31		10.56	47.96*
	Consultor	170.75			58.52*
	Jefe de Proyecto	229.27			
Objetivos asignados	Programador	150.73	20.42	19.72	74.34*
	Analista	171.15		0.70	53.92*
	Consultor	170.45			54.62*
	Jefe de Proyecto	225.08			
Software	Programador	218.53	69.03*	76.27*	69.85*
	Analista	149.49		7.24	0.82
	Consultor	142.25			6.42
	Jefe de Proyecto	148.68			
Hardware	Programador	209.18	50.11*	51.75*	72.66*
	Analista	159.08		1.65	22.55
	Consultor	157.43			20.91

			Diferencias observadas		
Ítem	Puesto de Trabajo	Rango Medio	Analista	Consultor	Jefe de Proyecto
	Jefe de Proyecto	136.53			
Especificación de requisitos	Programador	157.02	44.70*	11.88	34.00
	Analista	201.72		56.58*	10.70
	Consultor	145.14			45.88
	Jefe de Proyecto	191.02			
Conocimiento funcional	Programador	153.57	42.66*	23.09	19.17
	Analista	196.23		19.57	23.50
	Consultor	176.66			3.93
	Jefe de Proyecto	172.73			
Cliente	Programador	132.28	46.73*	86.15*	66.80*
	Analista	179.02		39.41	20.07
	Consultor	218.43			19.35
	Jefe de Proyecto	199.08			
Motivación	Programador	160.33	1.96	34.43	32.46
	Analista	162.29		32.47	30.49
	Consultor	194.76			1.98
	Jefe de Proyecto	192.78			
Código fuente previo	Programador	224.73	60.71*	95.67*	97.68*
	Analista	164.02		34.96	36.97
	Consultor	129.06			2.01
	Jefe de Proyecto	127.05			

Tabla 28. Rangos medios y diferencias observadas para las entradas (Kruskal Wallis, con $p < 0.05$)

* Hay diferencias significativas entre estos grupos

De otra parte, el puesto de trabajo con más diferencias en el grado de producción de las salidas empleadas es el Programador con 22 diferencias encontradas seguido del Jefe de Proyecto con 9 (de 96 posibles diferencias). El mismo patrón encontrado en las entradas está presente en las salidas. Por ejemplo, la producción de *Código fuente*,

Planificación, Ventas, y Solución de bugs son estadísticamente diferentes desde el resto de punto de vista de los otros puestos de trabajo con respecto al programador. Estos patrones añaden apoyo adicional al contraste de las hipótesis H2 y H4 ya que indican que la diferencia es mayor entre este puesto y los demás. Así pues, es posible decir que el puesto de trabajo Programador es el puesto con más diferencias encontradas en el estudio, seguido del puesto de Jefe de Proyecto.

Ítem	Puesto de Trabajo	Rango Medio	Diferencias observadas		
			Analista	Consultor	Jefe de Proyecto
Código fuente	Programador	244.85	102.68*	120.22*	120.31*
	Analista	142.16		17.53	17.62
	Consultor	124.63			0.09
	Jefe de Proyecto	124.54			
Producto	Programador	203.67	41.89*	66.80*	37.68
	Analista	161.78		24.91	4.21
	Consultor	136.88			29.11
	Jefe de Proyecto	165.99			
Documentación	Programador	139.54	62.42*	48.92*	40.57
	Analista	201.96		13.50	21.85
	Consultor	188.46			8.35
	Jefe de Proyecto	180.11			
Cumplimiento de objetivos	Programador	157.94	15.46	17.21	43.47*
	Analista	173.40		1.75	28.01
	Consultor	175.15			26.25
	Jefe de Proyecto	201.41			
Estimación	Programador	136.19	51.99*	38.80	87.31*
	Analista	188.18		13.19	35.32
	Consultor	174.99			48.51
	Jefe de Proyecto	223.50			
Planificación	Programador	129.21	63.99*	40.86*	106.20*
	Analista	193.20		23.12	42.21

			Diferencias observadas		
Ítem	Puesto de Trabajo	Rango Medio	Analista	Consultor	Jefe de Proyecto
	Consultor	170.08			65.33*
	Jefe de Proyecto	235.41			
Ventas	Programador	135.16	40.00*	73.10*	75.06*
	Analista	175.16		33.10	35.06
	Consultor	208.26			1.96
	Jefe de Proyecto	210.22			
Pruebas	Programador	202.54	35.40	69.74*	38.29
	Analista	167.15		34.34	2.89
	Consultor	132.81			31.45
	Jefe de Proyecto	164.26			
Solución de bugs	Programador	225.38	73.11*	96.29*	81.14*
	Analista	152.28		23.19	8.04
	Consultor	129.09			15.15
	Jefe de Proyecto	144.24			
Satisfacción del cliente	Programador	143.14	25.10	58.18*	68.92*
	Analista	168.24		33.08	43.82*
	Consultor	201.32			10.74
	Jefe de Proyecto	212.06			

Tabla 29. Rangos medios y diferencias observadas para las salidas (con $p < 0.05$ en Kruskal Wallis)

* Hay diferencias significativas entre estos grupos

3.4.5. Situación tras analizar los resultados

Los resultados pueden ser discutidos desde varios puntos de vista. Desde el punto de vista de la definición, la norma IEEE 1045-1992 define la productividad como la relación entre una primitiva de salida y su correspondiente primitiva de entrada para desarrollar software. Así, a partir de los resultados es posible establecer varias relaciones entre entradas y salidas. Por ejemplo, es posible establecer una relación

entre el código fuente desarrollado y el conocimiento utilizado para producirlo, o entre el código fuente previo y los bugs solucionados, o entre el conocimiento utilizado y el conocimiento generado. Sin embargo, parece que esta definición no cubre todas las posibles relaciones porque establece una relación entre una salida y una entrada y varios de estos elementos no están relacionados en pares, por ejemplo el hardware utilizado (entrada) y las ventas (salidas). Además, esta definición está reducida a una relación entre entradas y salidas (relación de tipo 1-1) y los resultados apuntan a relaciones múltiples entre entradas y salidas (relación de tipo M-N).

Desde el punto de vista de las medidas más utilizadas en IS, los resultados son controvertidos. Al igual que en el caso de la definición dada por la norma IEEE 1045-1992, las medidas más utilizadas se basan en un ratio (relación) entre una medida de tamaño de producto (SLOC o PF) y el esfuerzo requerido para producir el producto (horas o horas-hombre). Sin embargo, los resultados indican que hay más de una entrada y salida, por lo que varias relaciones podrían ser establecidas. Esta limitación en la definición de la productividad y en las medidas utilizadas pueden ser explicada en parte por el nivel de medición en el que se crearon: a nivel de proyecto. Pero si la definición de productividad no es la misma en todos los niveles (Hernández-López, Colomo-Palacios, & García-Crespo, 2012), utilizar la misma medida de productividad en todos los niveles no parece tener sentido.

Además, las medidas más utilizadas no son medidas eficaces para medir la productividad de estos profesionales a la luz de los resultados obtenidos. Estas medidas en cualquier caso deben ser consideradas como medidas específicas de productividad. Por ejemplo, la medida PF/t tiene que ser vista como una medida de productividad de entrega de software, es decir, una medida en la cual se mide la salida por la funcionalidad y sólo se requiere tiempo para producir la funcionalidad. Estas medidas además dejan fuera otras salidas y entradas a pesar de que varias de ellas podrían incluirse, teniendo en cuenta que en niveles inferiores de medición la granularidad puede ser reducida.

A su vez, si se utiliza una medida que sólo tiene en cuenta una salida y una entrada, cuando se busque una mejora de la productividad, cualquier modificación en otras entradas y salidas será un factor externo y no será incluido en la medida. Por ejemplo, si una organización da una formación adicional a los jefes de proyecto para mejorar la planificación y estimación de los proyectos (esto es una entrada de tipo formación) con el objetivo de mejorar la productividad de la entrega de software, es decir, incrementar la salida por unidad de esfuerzo, no habrá forma de comprobar si los resultados de productividad posteriores a la formación están influenciados por la formación recibida porque ésta es un factor, pero no es el único factor que se modificará en las siguientes mediciones. En este ejemplo, quizá una medida de productividad adicional para medir la productividad de la planificación y la estimación podría arrojar luz en conjunto con otras medidas de productividad. Para esto se podría emplear el método *Data Envelopment Analysis* (DEA) como ha sido introducido previamente para fines similares (Mahmood, et al., 1996).

Dentro de los resultados hay varias entradas que fueron mencionadas como salidas también, por ejemplo el conocimiento y la experiencia. Esta característica añade una dificultad extra a la tarea de medir la productividad de los profesionales de IS. Este hecho introduce un nuevo concepto a la definición de productividad y las medidas comúnmente utilizadas, se trata de la transformación de entradas en salidas del mismo tipo. En las medidas y definiciones convencionales, las entradas y las salidas son compartimentos aislados, es decir las entradas son sólo salidas y las salidas sólo salidas. Sin embargo, los resultados indican la existencia de varias entradas que son a su vez salidas. Estas entradas se modifican normalmente a la vez que el profesional desempeña las funciones de su puesto de trabajo.

Es ampliamente aceptado que si un trabajador entra y/o sale de un proyecto empezado, los resultados de la productividad cambiarán. Pero, si el tiempo y el esfuerzo son las únicas entradas utilizadas por los profesionales en IS para desarrollar software, tal y como se concluye al analizar las medidas más utilizadas, ¿cómo es

posible que la productividad cambie cuando la composición del equipo del proyecto cambia? Una de las posibles respuestas es que hay otras entradas que no son medidas en las medidas empleadas. Si sólo se utiliza el tiempo o el esfuerzo a la hora de medir la productividad, entonces los cambios en la productividad no serán significantes cuando la composición del equipo del proyecto cambie. En otras palabras, los puestos de trabajo en IS no son automatizables ni estandarizables como lo son en la industria manufacturera (Davenport & Prusak, 2000).

Llegado este punto, disponer de una descripción de los puestos de trabajo parece un elemento importante para medir (y mejorar) la productividad. Las definiciones de puestos de trabajo incluyen una lista completa de entradas utilizadas y de salidas producidas junto con los procesos de trabajo que se realizan (Fernández-Ríos, 1995). Así, si hay diferencias entre las entradas utilizadas (H2) y entre las salidas producidas (H4) para cada puesto de trabajo, y las medidas de productividad son específicas para cada puesto de trabajo, se debería tener en cuenta las entradas utilizadas y las salidas producidas para cada puesto como punto de partida. Esta tarea puede ser realizada mediante un análisis y descripción de puestos de trabajo. Finalmente hay que concluir que carece de sentido utilizar las mismas medidas de productividad para todos los puestos analizados si existen diferencias en las entradas a utilizar y en las salidas a producir entre los puestos. En resumen, a la hora de desarrollar las futuras medidas de productividad se debería considerar como guía base las definiciones de puestos de trabajo.

3.4.6. Validez y fiabilidad

Debido a la forma en la que se seleccionó a los participantes, es posible que exista un sesgo de selección producido por la autoselección. Los participantes fueron autoseleccionados cuando recibieron el email (esto sucede en ambas fases de la investigación) o cuando leyeron el post en LinkedIn (segunda fase), es decir, ellos mismos decidieron participar en la investigación. Este sesgo compromete la validez

externa de los resultados obtenidos. Este tipo de sesgo está presente en ambas fases de la investigación y debería ser solucionado en futuras investigaciones mediante el uso de un método de selección de participantes en el que no exista auto participación. Sin embargo, debido a la naturaleza de las hipótesis planteadas el efecto de este sesgo no debería afectar a los resultados obtenidos hasta el momento, ya que se trata de hipótesis exploratorias en las cuales no hay una relación estricta entre las variables bajo estudio. Por otro lado, podría replicarse el estudio con otros participantes para comprobar si los resultados son los mismos, utilizando las mismas entradas y salidas u otras, y/o añadiendo o borrando elementos si se considerase necesario.

3.5. Situación tras ambas fases de estudio

Los resultados hasta el momento indican que se utilizan otras entradas a parte del tiempo y el esfuerzo (H1), y se producen otras salidas a parte de la funcionalidad y el código fuente (H3) en los puestos de trabajo relacionados con los proyectos de desarrollo de software. Estos hallazgos ponen en entredicho la eficacia de las medidas de productividad más utilizadas en las que se mide la relación entre el tamaño del producto y el tiempo o esfuerzo empleado en producir el producto. Así, esas medidas deben ser consideradas como medidas específicas de productividad (Petersen, 2011), por ejemplo SLOC/t como productividad de entrega de código fuente, y su eficacia para medir un concepto más global como es la productividad está comprometido. Además, al haberse encontrado diferencias entre el uso de varias entradas a nivel de puesto (H2) y en la producción de varias salidas (H4) entre algunos de los puestos de trabajo, las medidas de productividad deberían tener en cuenta las características de cada puesto, es decir, las medidas deberían estar ajustadas a cada puesto de trabajo si el objetivo que se persigue con ellas es medir la productividad de dichos puestos.

De forma general, y teniendo en cuenta los resultados, los puestos de trabajo bajo estudio pueden ser agrupados dentro de los trabajadores del conocimiento. Estos puestos producen otras salidas, algunas de ellas no tangibles, que no son

tradicionalmente medidas como elementos producidos (p. ej., la experiencia y el conocimiento), utilizan otras entradas que no son *recursos humanos* per se (p. ej., la formación y la documentación), e interactúan con otros trabajadores y personas (p. ej., con el cliente para la toma de requisitos, y con otros compañeros) para desempeñar las tareas propias de su puesto de trabajo. Además, la calidad es una salida para estos puestos de trabajo. Por ello, de estas conclusiones se argumenta que emplear la misma filosofía que en el sector de manufactura para medir la productividad en los puestos de trabajo con fuerte carga de capital humano carece de sentido.

Tras realizar ambas fases, las hipótesis de la tesis doctoral se encuentran en la situación que se resume en la Tabla 30. En concreto, ha sido posible confirmar las cuatro primeras hipótesis, en las que se planteaba: la existencia de otras entradas además del tiempo y el esfuerzo (H1), la existencia de otras salidas además del código fuente y la funcionalidad (H3), la existencia de diferencias en el uso de las entradas en función del puesto de trabajo (H2), y la existencia de diferencias en la producción de las salidas en función del puesto de trabajo (H4). Además, en la fase cualitativa se obtuvo como resultado preliminar para H5 la invalidez de las medidas basadas en SLOC o PF para medir la productividad de los trabajadores bajo estudio de forma genérica y con la necesidad de ajustes en el caso de las basadas en PF. Y, en la fase cuantitativa, la existencia de múltiples entradas y múltiples salidas, junto con las diferencias en el uso y producción de las mismas en función del puesto contrasta preliminarmente esta hipótesis. Así pues, las medidas basadas en PF y/o SLOC como unidades exclusivas de salida, y tiempo y/o esfuerzo como unidades exclusivas de entrada, parecen no ser eficaces para medir la productividad de los puestos de trabajo bajo estudio. Además, es necesario utilizar medidas que consideren múltiples entradas ya que se utilizan otras entradas además del tiempo y el esfuerzo, y múltiples salidas ya que se producen otras salidas además del código fuente y la funcionalidad. Para contrastar las hipótesis H5 y H6, se plantea a continuación un estudio empírico en el que se desarrollaran nuevas medidas de productividad teniendo en cuenta los

resultados obtenidos y el conocimiento obtenido de estudio del estado de la cuestión realizado previamente.

Hipótesis	Contrastada	Resultado
Hipótesis 1. En los puestos de trabajo involucrados en la ejecución de proyectos de desarrollo de software se emplean otras entradas, además del tiempo y el esfuerzo.	Sí, desde un punto de vista exploratorio y estadístico.	Existen otras entradas además del tiempo (ver Tabla 22).
Hipótesis 2. Las entradas utilizadas son distintas para cada puesto de trabajo involucrado en la ejecución de proyectos de desarrollo de software.	Sí, desde un punto de vista estadístico.	Hay diferencias en el uso de varias de las entradas utilizadas en la fase cuantitativa para los cuatro puestos de trabajo bajo estudio, por lo que es posible afirmar que las entradas son distintas para cada puesto de trabajo en los proyectos de desarrollo software (ver Tabla 26 y Tabla 28)
Hipótesis 3. En los puestos de trabajo involucrados en la ejecución de proyectos de desarrollo de software se producen otras salidas, además de líneas de código y la funcionalidad.	Sí, desde un punto de vista exploratorio y estadístico.	Existen otras entradas además del código fuente y la funcionalidad (ver Tabla 23).
Hipótesis 4. Las salidas producidas son distintas para cada puesto de trabajo involucrado en la ejecución de proyectos de desarrollo de software.	Sí, desde un punto de vista estadístico.	Hay diferencias en la producción de varias de las salidas utilizadas en la fase cuantitativa para los cuatro puestos de trabajo bajo estudio, por lo que es posible afirmar que las salidas son

Hipótesis	Contrastada	Resultado
		distintas para cada puesto de trabajo en los proyectos de desarrollo software (ver Tabla 26 y Tabla 28).
Hipótesis 5. Las actuales medidas de productividad a nivel de puesto de trabajo en los proyectos de desarrollo de software tienen una eficacia limitada para medir la productividad real de los trabajadores.	Sí, desde un punto de vista exploratorio. Aunque parcialmente se ha contrastado esta hipótesis en la fase cualitativa, será contrastada tras el estudio empírico.	No, previamente a la realización de la validación, tiene sentido suponer que las medidas basadas en SLOC o PF por una unidad de esfuerzo o tiempo no son eficaces para medir la productividad de todos los puestos.
Hipótesis 6. Es posible medir de forma más eficaz la productividad de los puestos de trabajo en los proyectos de desarrollo de software con nuevas medidas que combinen varios elementos: entradas, salidas y factores.	No, será contrastada tras el estudio empírico.	Pendiente.

Tabla 30. Estado de las hipótesis tras ejecutar la fase cuantitativa

4 Solución propuesta

4.1. Introducción

Tras el estudio de la literatura mediante un análisis clásico y una posterior SLR, ha sido posible constatar que existen pocos estudios acerca de la medición de la productividad a nivel de puesto de trabajo en los proyectos de desarrollo de software (Petersen, 2011). Además, se ha obtenido el conocimiento necesario para diseñar y ejecutar el enfoque del problema, mediante una metodología mixta que ha incluido una fase cualitativa y una posterior cuantitativa. Este enfoque ha tenido como objetivos principales contrastar las hipótesis planteadas y profundizar en el conocimiento del objeto de la investigación. El contraste de las hipótesis ha sido confirmatorio de modo que en los puestos de trabajo vinculados con los proyectos de desarrollo de software: (H1) se emplean otras entradas – recursos – además del tiempo y el esfuerzo, (H2) se producen otras salidas – productos y servicios – además del código fuente y la funcionalidad, (H3) la utilización de las entradas varía en función del puesto y (H4) la producción de las salidas varía en función del puesto. Además, dado que se confirman las cuatro primeras hipótesis, de forma preliminar es posible confirmar que (H5) la utilización de las medidas más empleadas para medir la productividad, a nivel de puesto de trabajo en los proyectos de desarrollo de software, tiene una eficacia limitada a la hora de. Con el objetivo de contrastar las dos últimas hipótesis planteadas, se propone la presente solución.

Las dos hipótesis que se pretende contrastar con la solución propuesta son las siguientes:

Hipótesis 5. *Las actuales medidas de productividad a nivel de puesto de trabajo en los proyectos de desarrollo de software tienen una eficacia limitada para medir la productividad real de los trabajadores.*

Hipótesis 6. *Es posible medir de forma más eficaz la productividad de los puestos de trabajo en los proyectos de desarrollo de software con nuevas medidas que combinen varios elementos: entradas, salidas y factores.*

Para llevar a cabo el contraste de estas dos hipótesis es necesario desarrollar nuevas medidas de productividad, y posteriormente comparar su eficacia con las habitualmente utilizadas. De entre las medidas disponibles para medir la productividad a nivel de puesto de trabajo (Petersen, 2011; Ramirez & Nembhard, 2004), se ha optado por la metodología llamada *Data Envelopment Analysis* (DEA) dado que cubre más dimensiones de productividad que otras disponibles para medir la productividad de trabajadores del conocimiento (Ramirez & Nembhard, 2004). Esta metodología tiene como principales ventajas (1) la posibilidad de utilizar múltiples entradas y salidas, y (2) la ausencia de una relación ponderada preestablecida entre las entradas y salidas. Dentro de la medición de la productividad de los proyectos de desarrollo del software pueden encontrarse que la utilizan varios estudios, según la revisión sistemática llevada a cabo por Petersen (2011). Por ejemplo ha sido aplicada a nivel de proyecto (Asmild, et al., 2006; Banker, et al., 1991; Mahmood, et al., 1996; Stensrud & Myrtveit, 2003; Yang & Paradi, 2009), tarea (Ruan, et al., 2007), y a nivel individual (Liping, et al., 2005). Este último (Liping, et al., 2005), que sería el más próximo a los objetivos de la presente tesis doctoral, tiene por objetivo explicar la eficiencia relativa a la implantación de *Personnel Software Process* (PSP) (Humphrey, 1997) comparando el rendimiento y capacidades de PSP en diferentes fases, utilizando

estudiantes como muestra. No obstante, aunque el nivel de aplicación de DEA fue similar, el propósito final difiere del planteado en esta tesis doctoral: la medición de la productividad en los puestos de trabajo relativos a la ejecución de proyectos de desarrollo de software.

4.2. Data Envelopment Analysis (DEA)

El método DEA fue creado de forma preliminar en los años 50 por Farrell (1957). Posteriormente Charnes, Cooper y Rhodes lo ampliaron a finales de los 70 (1978) utilizando rendimientos constantes a escala, y Banker, Charnes y Cooper (1984) considerando rendimientos variables a escala, lo que permite calcular eficiencias de escala¹⁶. DEA es un método de programación lineal no paramétrico que no fija una forma explícita sobre los datos y crea una función de producción mediante la inclusión en el análisis de varias categorías de entradas (recursos) y salidas (productos o servicios). Para ello utiliza unidades de análisis denominadas *Decision Making Unit* (DMU). Una DMU debe completar tipos de actividad similares, producir tipos de productos y servicios similares, consumir tipos de recursos similares, y desempeñarse bajo restricciones ambientales similares (Farris, Groesbeck, Van Aken, & Letens, 2006). Farrell propuso que la eficiencia de una DMU consiste en dos componentes: “eficiencia técnica”, que refleja la habilidad para obtener el máximo output para un conjunto dado de inputs, y la “eficiencia redistributiva”, que refleja la habilidad para usar los inputs en las proporciones óptimas, dados sus respectivos precios. Ambas medidas se combinan para obtener la “eficiencia económica”.

16 Los rendimientos a escala se refieren a una propiedad técnica de la producción que analiza los cambios en las salidas paralelo y proporcional en todas las entradas (cuando todas las entradas incrementan por una constante). Si la salida aumenta en la misma proporción que las entradas, entonces existe retornos a escala constante, a veces mencionada simplemente como rendimientos a escala. Si la salida aumenta por menos que el cambio proporcional en las entradas, hay rendimiento a escala decreciente; y si aumenta por más que el cambio entonces hay rendimiento a escala creciente. En estos dos casos, los rendimientos a escala son de tipo variable.

Este método compara cada medición con el resto de mediciones, y por lo tanto no se tiene un resultado universal. El propósito del DEA es hacer que el valor de productividad para cada DMU, en la muestra, sea el máximo que pueda alcanzar; para ello, se ajustan los pesos de la combinación de variables de entrada y de salida, de acuerdo con el resto de las DMU de la muestra. Además, el método DEA tiene modelos que pueden ser clasificados como radiales orientados a las entradas, radiales orientados a las salidas, o aditivos, donde tanto las entradas como las salidas son optimizadas basadas en la dirección de la proyección de la unidad ineficiente hasta la frontera. Para una detallada información sobre los modelos orientados a entradas y salidas, consultar (Charnes, Cooper, Lewin, & Seiford, 1994).

El crecimiento de la literatura científica que utiliza DEA en alguna de sus variantes es de carácter exponencial y parece que las modificaciones realizadas sobre los modelos iniciales van a continuar en los próximos años (J. S. Liu, Lu, Lu, & Lin, 2013). Para conocer de forma resumida las modificaciones más utilizadas, ver el artículo realizado por Cook y Seiford (2009).

Así pues, la idea básica de aplicación de DEA dentro de esta tesis doctoral es la construcción de un trabajador modelo dentro de cada puesto de trabajo conformado por la combinación de las entradas y salidas de todos los trabajadores analizados, y la identificación de la denominada frontera de eficiencia. Todos los trabajadores que estén en la frontera serán aquellos que estén produciendo al cien por cien de eficiencia para las variables de entrada y salida seleccionadas. Los trabajadores que estén fuera de la frontera de eficiencia, serán los trabajadores *improductivos* pudiéndose calcular el valor relativo de esta *improducción*.

Con la finalidad de formalizar el modelo, consideramos la función objetivo del problema:

Dada la DMU_j , el objetivo es:

donde,

y_{rj} es el valor de la variable de salida r en la DMU j -ésima para $r = 1, \dots, s$;

x_{ij} es el valor de la variable de entrada i en la DMU j -ésima para $i = 1, \dots, m$;

u_r es el peso de la variable de salida r -ésima;

v_i es el peso de la variable de entrada i -ésima y

n es el número de DMUs.

El objetivo, como hemos señalado anteriormente, es maximizar los pesos u y v para que el valor de productividad resultante para esa unidad de análisis sea el mayor posible.

Las restricciones a este modelo son las siguientes:

- Los valores de productividad están acotados entre 0 y 1 (ya que son números relativos) y, por tanto:
-

- Los pesos que se utilicen deben ser valores no negativos y, por tanto:

Consecuentemente, el valor de productividad de la DMU _{j} se puede obtener resolviendo el siguiente modelo de programación lineal:

Las DMU con el valor máximo de productividad serán aquellas cuyo E_j alcance el valor de 1 y se las denomina unidades productivas, las cuales constituirán el conjunto de referencia para las unidades improductivas. La forma en que se construye la

frontera de eficiencia, a través de las variables de entrada y salida, determina que no se puedan incluir en el análisis un número indeterminado de variables, dado que cada vez que se incorpora en el modelo una nueva variable es de esperar que se obtenga una nueva DMU eficiente, siempre y cuando la variable introducida no sea redundante en el modelo. El producto del número de variables de entrada y de salida no debe superar el número de unidades que conformen la muestra del estudio (Boussofiane, Dyson, & Thanassoulis, 1991); en este caso, el producto de las variables de entrada y de salida no debe superar el número de trabajadores de los que deseamos calcular su productividad. En caso contrario, el resultado puede conducir a que una gran proporción de los trabajadores sean productivos, reduciéndose la capacidad de discriminación de este método.

4.2.1. Características de los modelos DEA

1. **Isotonicidad.** Una regla fundamental para utilizar DEA es que un incremento en una variable de entrada debe mejorar las salidas; esto se llama isotonicidad de los parámetros DEA (Chung, Lee, Kang, & Lai, 2008). Para analizar esta característica es necesario llevar a cabo un análisis de correlación para asegurar relaciones positivas entre las entradas y las salidas. En el caso de encontrar correlaciones negativas, se pueden aplicar técnicas de transformación. Sin embargo, varios problemas relacionados con la transformación de variables para satisfacer la isotonicidad han sido descritas en la literatura (W. Liu, Meng, Li, & Zhang, 2010). Así pues, en lugar de realizar transformaciones de parámetros, es mejor plantearse las correlaciones negativas como un indicador que señala que uno o más parámetros deben ser excluidos del modelo.
2. **Orientación del modelo DEA.** Existen dos orientaciones básicas en los modelos DEA: maximización de salidas y minimización de entradas (Cook & Seiford, 2009). Un modelo de maximización de salidas indica el máximo incremento en la proporción de salidas respecto de las entradas, lo que es adecuado en situaciones en las cuales se busca un objetivo de salida. Por otro

lado, un modelo de minimización de entradas indica la cantidad mediante la cual las entradas pueden ser reducidas mientras que los valores de salida son los mismos.

3. **Selección del modelo DEA.** En la selección del modelo DEA es necesario asumir si existen o no rendimientos a escala. Los trabajadores en IS posiblemente trabajen bajo escalas positivas y negativas, por lo que, salvo que esté claramente identificado el rendimiento, parece apropiado seleccionar un modelo que permita a la frontera de eficiencia operar bajo rendimientos negativos a escala; para ello se puede utilizar un modelo que asuma rendimientos a escala variables (Otero, Centeno, Otero, & Reeves, 2012).
4. **Homogeneidad de DMUs.** Los modelos DEA asumen que el conjunto de DMUs es homogéneo para que las comparaciones entre ellas sean en igualdad de condiciones. Para cumplir esta necesidad cuando se está evaluando productividad da nivel de puesto de trabajo es necesario considerar los parámetros de entorno, incluidos el entorno de trabajo y cualquier variable que pueda afectar de forma significativa a los valores de las DMUs. Para ello, es posible tratar datos sólo de una organización (mismo entorno de trabajo, misma cultura...) o clasificar las DMUs de acuerdo a determinadas condiciones que garanticen la homogeneidad de las mismas (Otero, et al., 2012).

4.2.1.1. Ventajas y desventajas de DEA

Las ventajas de DEA son los siguientes (Cooper, Seiford, & Tone, 2006):

- No necesita una forma matemática específica y explícita para la función de producción.
- Se ha contrastado como un método útil en relaciones no cubiertas y que permanecen ocultas para otras metodologías.
- Es capaz de tratar con múltiples entradas y salidas.
- Es capaz de ser utilizado con cualquier medida de entrada-salida.

- Las fuentes de improductividad pueden ser analizadas y cuantificadas para cada unidad evaluada. Para ello, podemos utilizar dos procedimientos:
 - Identificar, para cada uno de los trabajadores improductivos, cuál o cuáles son sus trabajadores de referencia, y así poder conocer las variables sobre las que se debe actuar para que consigan alcanzar la productividad. Para ello, se pueden utilizar dos procedimientos:
 - Realizar un análisis global de sensibilidad que permita identificar, para cada uno de los trabajadores improductivos, en cuánto deberían incrementar sus variables de salida, o en cuánto deberían reducir los recursos de entrada con el fin de alcanzar la frontera de eficiencia. Para obtener estos datos es posible emplear la modificación propuesta por Chen (1997) del método de Charnes, Cooper y Rhodes (1978).
- Es superior a la regresión estadística en las situaciones en las que existen múltiples entradas y múltiples salidas sin una forma objetiva de agregar los resultados en un índice de producción con significado.
- Evalúa la productividad de una DMU de forma relativa al resto de las productividades de las DMUs, por lo que siempre se compara con el mejor en lugar de ser comparado con la media, tal y como sucede con los métodos de regresión lineal.

Las desventajas de DEA que han sido documentadas (Cooper, et al., 2006) son:

- Los modelos DEA pueden ser sensibles a valores atípicos (ver p. ej.: Liping, et al., 2005) se produce esa situación; pero no siempre se produce esta desventaja (ver p. ej.: Banker, et al., 1991; Ruan, et al., 2007; Stensrud & Myrtveit, 2003). En caso de producirse un efecto negativo la frontera de eficiencia cambia, y por lo tanto algunos de las DMU serán marcadas como eficientes, poniendo en duda el resultado obtenido para el resto de DMUs. No obstante, la existencia de valores atípicos no es tan problemática como

en el caso del uso de regresiones lineales, o de simples ratios (salida/entrada).

- Los datos obtenidos en los estudios IS pueden haberse obtenido sin tener en cuenta las premisas de aplicabilidad de DEA.
- Los errores de medición pueden originar diferencias importantes en los resultados.
- Dado que DEA calcula la eficiencia relativa, no es posible obtener la eficiencia absoluta; es decir, es imposible conocer el valor máximo teórico.
- Hace años, cuando la capacidad de computación era limitada, la resolución de modelos DEA con un alto número de variables y/o DMUs, suponía un problema. Actualmente esta desventaja no se debe tener en cuenta dada la alta capacidad de los computadores, en comparación con la capacidad que tenían hace 20 años.

4.2.2. DEA en IS

En IS, el método DEA ha sido utilizado con múltiples objetivos, todos ellos relacionados con la producción de software en mayor o menor medida, desde finales del siglo pasado. Por ejemplo, ha sido utilizado con diversos objetivos de investigación, entre ellos se pueden mencionar los siguientes: para analizar los factores que influyen el tamaño de los componentes en Orientación a Objetos y de la documentación del código fuente (Pendharkar, 2006), para clasificar varias versiones de un software basándose en los valores de métricas de diseño y características de tamaño (Chatzigeorgiou & Stiakakis, 2013), para comparar y mejorar sistemas basados en conocimiento (Mei-Chi, Hao-Chen, & Wei-Kang, 2011), para analizar los rendimientos a escala en el desarrollo de software (Banker & Kemerer, 1989), para analizar los drivers de rendimiento de proyectos de software libre, tanto desde el punto de vista de la eficiencia como de la efectividad (Ghapanchi & Aurum, 2012), para evaluar la relación entre los PF post implementación y el esfuerzo de desarrollo correspondiente (Asmild, et al., 2006), para analizar la relación entre el tamaño del equipo y el coste

del desarrollo de software (Pendharkar & Rodger, 2009), para identificar proyectos con alto rendimiento (Stensrud & Myrtveit, 2003), para medir la productividad de varios proyectos (Mahmood, et al., 1996), para comparar la productividad de proyectos COTS (Myrtveit & Stensrud, 1999), para evaluar el rendimiento de proyectos software (Yang & Paradi, 2009; B. Yu, 2010) y el rendimiento de proyectos de diseño (Farris, et al., 2006), y para evaluar la capacidad de los procesos y prácticas de software individuales (Zhang, Wang, Yang, & Xiao, 2008). No obstante, Kitchenham (2004) ha planteado algunas objeciones a la aplicación de DEA a proyectos software. Según ella, los modelos DEA incluyen unos supuestos que no son siempre validos en este entorno. En concreto, los modelos DEA asumen que no hay errores de medición, que todas las variables de entrada están incluidas en el modelo, y que cada dato representa una DMU independiente.

4.2.3. DEA para medir la productividad de los trabajadores del conocimiento

El método DEA cubre cinco de las 13 dimensiones destacadas por Ramírez y Nembhard en su estudio sobre la medición de la productividad para los trabajadores del conocimiento (2004); mencionar que el método empírico que mas dimensiones cubre abarca cinco, mientras que dos trabajos teóricos cubren seis. En concreto el método DEA cubre: Cantidad, Coste y/o Rentabilidad, Eficiencia, Efectividad, y Éxito del Proyecto. En comparación, medir la productividad mediante PF solo cubre dos dimensiones, Cantidad y Calidad, según Ramírez y Nembhard (2004), por lo que medir la productividad mediante DEA es, al menos basado en estas dimensiones, más completo.

A nivel de trabajadores, ha sido utilizado para examinar la productividad, eficiencia y eficacia de un equipo de diseño en ingeniería (Paradi, et al., 2002), de manera conjunta con triangulación de lógica difusa para medir la productividad de trabajadores del conocimiento (Abdoli, Shahrabi, & Heidary, 2011), para optimizar la

asignación de recursos humanos a múltiples proyectos (Weng, Su, Chen, & Wang, 2010), para comprender el role de la experiencia y los factores de las tareas en la eficiencia de los ingenieros software (Otero, et al., 2012).

4.3. Solución propuesta basada en DEA

Teniendo en cuenta los resultados previos y los requisitos de aplicabilidad de DEA se plantea una solución configurable en función de las necesidades de medición de cada organización participante en la fase de validación. Para llevar a cabo esta solución, es necesario ejecutar las tareas incluidas en la Ilustración 19. En concreto, la solución permite la configuración de los siguientes parámetros:

1. Los **puestos de trabajo** que serán objeto de mediciones en cada estudio.
2. Para cada puesto de trabajo:
 - a. Las **variables de entrada** a medir.
 - b. Las **variables de salida** a medir.
 - c. Los **factores** a medir.

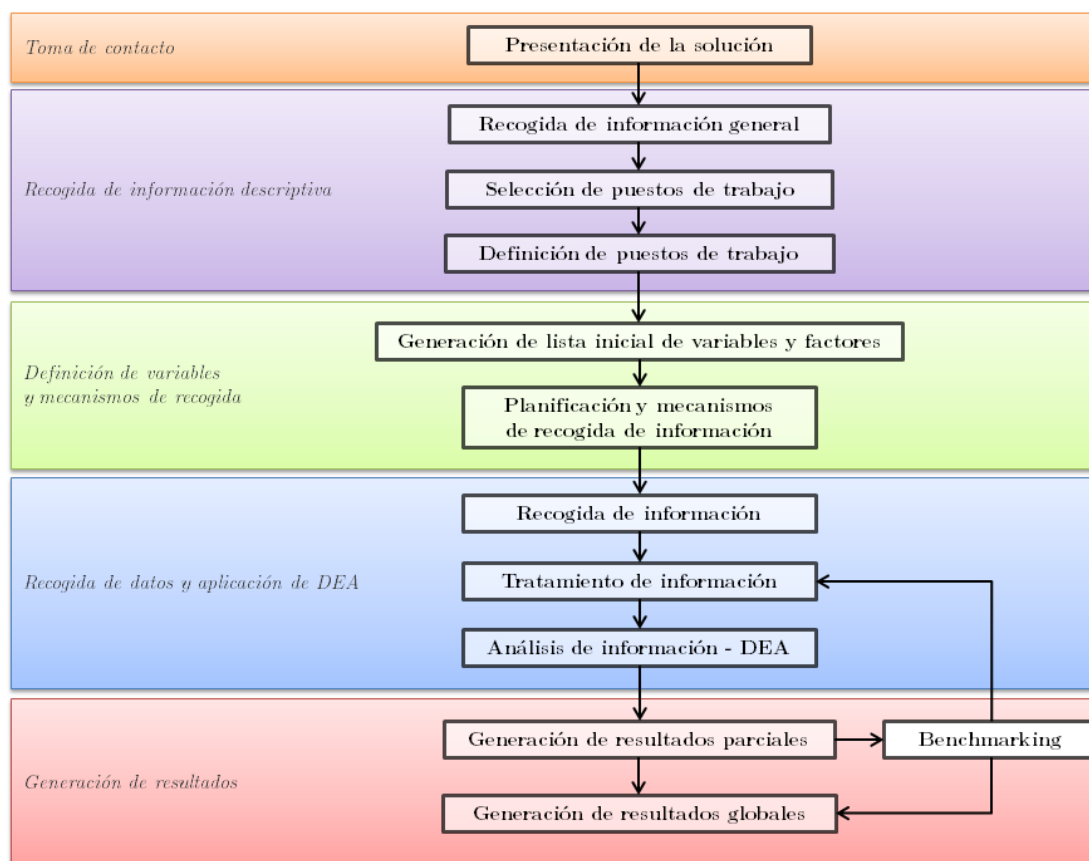


Ilustración 19. Solución propuesta basada en DEA

Las tareas de esta solución tienen los siguientes objetivos:

1. **Presentación de la solución.** El propósito de esta tarea es presentar, a cada organización la propuesta, los objetivos y fases de la misma, así como el esfuerzo necesario para llevar a cabo la solución propuesta.
2. **Recogida inicial de información de carácter general.** Una vez que una organización decide participar, se recoge información suficiente para describir la organización mediante una o varias entrevistas semiestructuradas con personas que ocupan puestos de gestión dentro de la organización (p. ej. jefes de proyecto o de departamentos). Esta información permite caracterizar a cada organización participante en la presente tesis doctoral, y en las publicaciones en las que se utilicen los datos obtenidos. Además, permite la publicación de los resultados de forma anónima gracias

a la disponibilidad de información suficiente para dar una descripción de la organización sin facilitar su nombre. Los elementos ha obtener son:

- a. Organización: tamaño y áreas de negocio.
 - b. Tipología de proyectos y metodología de trabajo: tipo, tamaño, tecnologías, planificación, y entregas a alto nivel de los proyectos más representativos.
 - c. Puestos de trabajo: lista de puestos de trabajo que participan en los proyectos de desarrollo de software.
 - d. Proceso de trabajo: proceso según el cual se llevan a cabo los proyectos de desarrollo de software dentro de la organización.
 - e. Herramientas, indicadores, y procedimientos de medición: lista de elementos relativos a la medición del proceso de trabajo disponibles, con especial interés en los relativos a la medición de la productividad, y del uso y producción de recursos y servicios.
3. **Selección de puestos objeto de estudio.** En base a la información obtenida en el punto anterior, es posible seleccionar los puestos de trabajo que podrían ser objeto de estudio. Esta decisión se basa en las siguientes condiciones (por este orden):
- a. *Número de personas que ocupan el puesto de trabajo.* Esto es, el número de DMUs disponibles para ser analizadas aplicando DEA. Dado que el método DEA realiza una comparación entre las DMUs, es necesario que haya un mínimo de personas ocupando un puesto de trabajo para tenga sentido realizar una comparación entre DMUs. Además, el número de DMUs condiciona el número de variables (de entrada y salida) que deben ser incluidas.
 - b. *Interés específico por parte de la organización en medir la productividad de un puesto de trabajo en concreto.* Si el número de puestos que cumplen el anterior requisito es superior a uno, se plantea a la organización la posibilidad de seleccionar un

subconjunto de puestos, con el fin de reducir el desempeño necesario para ejecutar la participación. Así pues, es la organización la que selecciona los puestos objeto de estudio para cumplir con esta condición.

4. **Definición de puestos de trabajo.** Esta tarea se realiza mediante un análisis de puestos de trabajo a través de entrevistas semiestructuradas realizadas a trabajadores clave dentro de la organización para cada uno de los puestos de trabajo. De esta tarea se obtienen los siguientes resultados:

- a. Para cada puesto de trabajo:
 - i. Definición de productividad a nivel de puesto de trabajo: a partir de esta definición es posible plantear una medida de productividad acorde al puesto de trabajo.
 - ii. Variables de entrada y salida potencialmente medibles: en cada puesto de trabajo se utiliza un grupo de recursos y se produce un grupo de salidas. Así pues, se obtienen las entradas y salidas potencialmente medibles.
 - iii. Factores que influyen en la productividad: de igual modo que para las entradas y salidas, se obtienen los factores incidentes en la productividad del puesto.
- b. Proceso de trabajo ajustado a la realidad: a partir de la información obtenida en las entrevistas es posible ajustar el proceso de trabajo, obtenido previamente, a la realidad. De este modo se genera un diagrama del proceso de trabajo llevado a cabo en los proyectos de desarrollo de software en el que se incluyen las tareas en las que participa cada puesto de trabajo así como las principales entradas y salidas propias del proceso productivo.

5. **Generación de la lista de variables y factores a medir.** A partir de la información obtenida en las definiciones de puestos de trabajo, y la información de carácter general, se genera una lista inicial de variables

(entradas, salidas, y factores) a medir para cada uno de los puestos de trabajo. En esta lista se incluye, para cada variable, los siguientes datos:

- a. Nombre: concepto que va a ser considerado en la medición.
- b. Frecuencia de medición: frecuencia con la que se realizará la medición.
- c. Procedencia: cómo se obtiene la medida de la variable.
- d. Persona encargada: en caso de que la medición no provenga de una aplicación de gestión, se indica quién es la persona que realiza la medición. Si por el contrario proviene de una aplicación de gestión, se indica el nombre de dicha aplicación.

6. **Planificación de medición y mecanismos de recogida de información.** El objetivo de esta tarea es realizar una planificación para llevar a cabo la medición y establecer los mecanismos de recogida de información que durante dicha planificación permiten la obtención de los resultados para las variables y factores a medir incluidos en la lista generada en el punto anterior.

7. **Recogida de información.** En esta tarea, mediante los mecanismos de recogida de información establecidos en el punto anterior, se recoge la información de las variables y factores.

8. **Tratamiento de información.** En esta tarea se comprueban los requisitos de aplicabilidad de DEA a partir de la información recogida. En concreto se realizan las siguientes comprobaciones/decisiones:

- a. Análisis de correlación entre variables del mismo tipo (entradas, salidas): el objetivo de esta comprobación es eliminar variables del mismo tipo que correlacionen entre sí. De este modo se obtiene un conjunto de variables reducido con el que llevar a cabo el análisis de la información mediante DEA.
- b. Análisis de isotonicidad: dado que DEA establece como requisito que el incremento de las variables de entrada debe incrementar las

variables de salida, se realiza un análisis de correlación para comprobar que existe una relación entre pares de entradas y salidas. En concreto, se descarta cualquier variable de entrada o salida que no esté correlacionada con alguna del grupo opuesto.

- c. Orientación del modelo DEA: en esta decisión se establece la orientación entre maximización de salidas o minimización de entradas en función de los objetivos de análisis marcados.
 - d. Selección del modelo DEA: en base a los retornos al escala supuesto se decide entre retornos a escala constantes o variables.
 - e. Homogeneidad de DMUs: con respecto al requisito de homogeneidad de las DMUs, al tratarse de personas que ocupan el mismo puesto de trabajo, dentro de la misma organización, esta comprobación debe considerarse como inicialmente cubierta. No obstante, si se detectan en los pasos previos una diferencia sustancial que ponga en riesgo este requisito, se puede clasificar por grupos las DMUs para dicho puesto en función del grupo de elementos que permitan caracterizar a cada subgrupo.
9. **Análisis de información.** Teniendo en cuenta las decisiones de la tarea anterior, se lleva a cabo el análisis de la información mediante un software específico para emplear la metodología DEA.
10. **Generación de resultados parciales.** A partir de los resultados obtenidos, es posible analizar los resultados para cada organización participante. En concreto es posible obtener los resultados de productividad, las DMUs de referencia y las variables sobre las que actuar para mejorar la productividad.
11. **Benchmarking con otras participaciones.** En esta tarea se lleva a cabo una comparativa entre los resultados obtenidos de cada una de las organizaciones participantes. Esta comparativa no es posible llevarla a cabo incluyendo todas las DMUs de cada una de ellas, dado que se incumpliría el

requisito de homogeneidad de DMUs, por lo que se realizará una comparativa de resultados a nivel de conclusiones. En caso de que existiera homogeneidad entre DMUs de distintas organizaciones, podría llevarse a cabo una comparativa contemplando la inclusión de las DMUs homogéneas en un nuevo análisis de información, es decir, volviendo a ejecutar las tareas 9 y 10.

12. **Generación de resultados globales.** En esta tarea se generan las conclusiones sobre la ejecución de la propuesta y se destacan los puntos a tener en cuenta para mejorar la propuesta.

5 Validación

5.1. Introducción

Siguiendo la solución propuesta basada en DEA detallada en la sección anterior, se ha buscado la colaboración de organizaciones que ejecutasen proyectos de desarrollo y/o mantenimiento de software para presentarles la propuesta. Conseguir organizaciones dispuestas a aportar datos para la investigación es una de las actividades más difíciles dentro de la investigación en IS (Agrawal & Chari, 2007). No obstante, pese a su dificultad, es una actividad necesaria ya que sin una validación fuera del entorno universitario, en el que en numerosas ocasiones se utilizan muestras de estudiantes sin justificación para tal selección, la validez de las soluciones que se proponen en las investigaciones es limitada. Para conseguir un conjunto de organizaciones participantes se ha seguido la siguiente aproximación desde Septiembre de 2012:

1. Se ha generado un listado de empresas de Madrid principalmente, aunque se localizaron algunas de otras provincias, cuya actividad incluyera la realización de proyectos de desarrollo y/o mantenimiento de software. En este listado se ha incluido a 103 empresas.

2. Se ha contactado vía email con estas organizaciones. Dado que algunas no disponían de email para realizar el envío sólo ha sido posible enviar 73 emails.
3. En caso de no obtener respuesta o no disponer de un email de contacto, se ha realizado una llamada a dicha organización para dar a conocer la solución propuesta y la colaboración solicitada para llevarla a cabo.
 - a. Los resultados para esta fase, y con 79 empresas, han sido:
 - i. 51 empresas han indicado un email para enviarles la propuesta y se continua a la espera de respuesta,
 - ii. 12 empresas eran mycropymes o autónomos por lo que su participación ha quedado descartada ya que en la mayoría de los casos no había varias personas ocupando un mismo puesto,
 - iii. 8 empresas no existen,
 - iv. 4 empresas han indicado que estaban interesadas pero tras enviarles la información de la propuesta no se ha obtenido respuesta,
 - v. y 4 empresas han contestado que no estaban interesadas en participar.

Dado que los resultados no habían sido satisfactorios hasta un Diciembre de 2012, se optó por no esperar respuestas de las organizaciones que dijeron estar interesadas y obtener la participación de una organización relevante para el ámbito de estudio. Además, esta colaboración con la citada empresa serviría como prueba piloto y permitiría refinar el proceso. No obstante, dicha colaboración no seguiría la propuesta de solución definida dado que: (1) partiría exclusivamente de información de carácter histórica disponible en una herramienta de gestión de bugs, y (2) la información sobre organización y proceso de trabajo se obtendría de las entrevistas realizadas en la fase cualitativa dado que varios de los participantes, en concreto un jefe de proyecto y tres ingenieros de software, fueron entrevistados en la dicha fase y la información necesaria

puede ser extraída de las mismas. Los resultados de este caso se encuentran en el apartado 5.2.1 Caso 1 - Prueba piloto.

Paralelamente se siguió la búsqueda de organizaciones potencialmente participantes por otros medios. En concreto, se presentó la propuesta a varios de los miembros asociados a la Asociación Española para la Gobernanza, la Gestión, y la Medición de las Tecnologías de la Información (AEMES) en la cual la Universidad Carlos III es asociada. Se obtuvo una respuesta afirmativa y se comenzó el proceso de colaboración con dicha organización. Los resultados de este caso se encuentran en el apartado 5.2.2 Caso de estudio 2.

Además, por otras vías se consiguió la participación de una organización multinacional de desarrollo de software que opera bajo distintos nombres comerciales. Así, se obtuvo la colaboración de dos sub-organizaciones internas a la marca internacional. Se dividió la colaboración en dos casos ya que, aunque existen ciertas áreas de la organización que son compartidas por ambas sub-organizaciones, la casuística relativa al proceso de trabajo, clientes, tipología de proyectos, organización del trabajo, etc., es totalmente distinta. Los resultados de estos dos casos se encuentran en los apartados 5.2.3 Caso de estudio 3 y 5.2.4 Caso de estudio 4.

5.2. Casos de estudio

La validación será realizada con cuatro casos de estudio, los cuales serán detallados a continuación. Para cada caso de estudio se describe la organización y el proceso de trabajo llevado a cabo para realizar los proyectos de desarrollo de software, se definen los puestos de trabajo, las variables y los factores que se analizan y miden, y se presentan los resultados de la preparación de la muestra en base a los requisitos de aplicación de DEA, se indican los resultados del análisis D, y finalmente los resultados parciales del caso de estudio en cuestión. Finalmente, se presentan las conclusiones globales obtenidas tras llevar los casos de estudio.

5.2.1. Caso 1 - Prueba piloto

5.2.1.1. Descripción de organización y proceso de trabajo

La organización participante puede describirse como una organización multinacional de desarrollo de software de gestión organizacional presente en más de 100 países. Sus productos son utilizados por 1.300 clientes. En la sede de España cuenta con algo menos de 500 trabajadores. En esta organización, el proceso de trabajo para llevar a cabo el desarrollo y mantenimiento de software es el siguiente:

1. Los analistas estudian, junto con el área de negocio, las posibles funcionalidades y cambios a incluir en futuras versiones de los productos, para posteriormente analizar las que finalmente serán incluidas.
2. Se establecen fechas de entrega de la versión, en las cuales se incluyen determinadas funcionalidades y cambios a incluir.
3. Los ingenieros de software reciben los análisis realizados por los analistas y llevan a cabo los diseños y desarrollos necesarios para que implementarlos.
4. Paralelamente al desarrollo, los ingenieros de software llevan a cabo revisiones y correcciones del producto en desarrollo así como de anteriores versiones del mismo, teniendo una carga de trabajo semanal asignada a dicha tarea.
5. Los ingenieros de software realizan pruebas detalladas de las funcionalidades desarrolladas antes de la entrega de la versión.

5.2.1.2. Puestos de trabajo, variables y factores a analizar

En este caso de estudio, se estudia la productividad del puesto de trabajo encargado de la resolución de bugs de un proyecto de desarrollo de software. Además de este objetivo, el puesto tiene otros tales como el desarrollo de nuevas

funcionalidades pero en el presente caso sólo se analiza la parte del puesto relativa a la resolución de bugs. El puesto de trabajo bajo estudio en este caso es el de Ingeniero Software dentro del área de I+D tecnológico de uno de los productos desarrollados por la organización participante. Este puesto diseña y desarrolla las funcionalidades que los analistas han analizado previamente, y que han sido designadas para ser incluidas en la siguiente versión del producto. Además, revisa y soluciona las anomalías encontradas en los productos, teniendo una carga semanal de trabajo asignada a esta labor.

Como variables de estudio se plantearon las siguientes:

	Nombre	Periodicidad	Escala
Entradas	Antigüedad en el puesto	1 vez, al principio	Entero (años)
	Experiencia tecnológica (.NET)	1 vez, al principio	Entero (años)
Salidas	Número de bugs solucionados	1 vez, al final	Entero
Factores	<i>No se plantearon factores</i>		

Tabla 31. Variables bajo estudio en el Caso 1

Se obtuvo información de los bugs solucionados durante los años 2010, 2011, y 2012.

5.2.1.3. Descripción demográfica de participantes

La muestra de 6 participantes, de los cuales cinco hombres (edad media 38,75, *de* 6,5), y una mujer (edad 37). Con respecto a la formación: cuatro disponen de Licenciatura (66.67%), y dos de Diplomatura (33,33%).

ID	Edad (<i>a 31/12/2012</i>)	Sexo	Estudios
1	36	Hombre	Licenciatura
2	30	Hombre	Diplomatura
3	38	Hombre	Licenciatura
4	37	Mujer	Licenciatura
5	45	Hombre	Diplomatura

6	42	Hombre	Licenciatura
---	----	--------	--------------

Tabla 32. Descripción demográfica de los participantes en el Caso 1

5.2.1.4. Preparación de datos

Los datos de las variables a analizar en este caso son los siguientes:

ID	Antigüedad puesto	Experiencia (.NET)	Bugs solucionados		
			2010	2011	2012
1	14	6	420	464	538
2	5	3	204	204	269
3	14	6	151	289	356
4	14	3	383	516	655
5	16	3	1238	1436	1469
6	16	3	462	556	551

Tabla 33. Participantes en el Caso 1

El primer paso para preparar los datos es llevar a cabo un estudio de correlación para comprobar por un lado la correlación entre variables del mismo grupo y por otro lado la isotonicidad. En este caso los resultados del coeficiente ρ (ro) de correlación de Spearman han sido los siguientes:

	Experiencia (.NET)	Antigüedad puesto
Número de Bugs	0,049 (Sig. 0,863)	0,805 (Sig. 0,000)
Experiencia (.NET)		0,256 (Sig. 0,358)

Tabla 34. Coeficientes de correlación (Caso 1)

Para llevar a cabo este análisis se han añadido combinaciones de variables (número de bugs, antigüedad y experiencia) para ajustar los valores de las entradas a todos los años. En concreto, la experiencia y antigüedad se han establecido en un año menos para las combinaciones de 2011, y dos menos para las de 2010, con respecto a los años indicados en la información demográfica de los participantes. Una vez realizados estos cambios, se ha llevado a cabo el análisis y la única correlación con significancia

encontrada es Antigüedad puesto y Número de Bugs a un nivel de confianza de 0,01. Así pues, la entrada Experiencia tecnológica, en este caso relativa a .NET, no está correlacionada con el número de bugs solucionado. De este modo, dado que sólo se encuentra una correlación de isotonicidad carece de sentido aplicar el método DEA, ya que obtendríamos los mismos resultados aplicando un ratio del tipo $Productividad = \text{Número de Bugs} / \text{Antigüedad puesto}$.

5.2.1.5. Generación y análisis de resultados

Dado que los datos obtenidos no cumplen los requisitos de aplicación de DEA (en concreto no cumple el requisito de isotonicidad) no es posible obtener resultados con dicho método.

5.2.1.6. Conclusiones parciales

De este caso de estudio se obtiene como conclusión general que la utilización de DEA requiere analizar un conjunto amplio de variables de forma inicial el cuál se verá reducido tras la preparación de los datos. En este caso, al tratarse de una aproximación basada en información histórica, la inclusión de un mayor número de variables ha resultado prácticamente imposible ya que la única variable de salida disponible ha sido el número de bugs. No obstante, el número de participantes en este caso ha sido muy limitado por lo que los coeficientes de correlación obtenidos pueden no ser representativos. Así pues, para los siguientes casos de estudio se ha de plantear una lista de variables más ampliada y un número mayor de participantes para minimizar la posibilidad de que se reproduzca esta situación.

Como conclusión específica hay que señalar que la antigüedad en el puesto es un recurso a tener en cuenta si se desea aumentar el número de bugs solucionados, dado que se ha encontrado una correlación positiva entre estas dos variables en los datos recibidos. Por el contrario, la antigüedad tecnológica, es decir, la experiencia con una tecnología, no presenta ninguna correlación. Esta falta de correlación puede explicarse

de la siguiente manera: los participantes han alcanzado un nivel a partir del cual no se produce un aumento de la salida analizada (bugs solucionados) y que por lo tanto no es necesaria más experiencia que la ya existente. Es decir, en este caso, la experiencia en la tecnología superior a tres años sería un recurso innecesario. No obstante, esto es explicación que no ha podido ser contrastada. Por ello, se plantea una posible futura línea de investigación que analice de forma aislada la influencia de diferentes tipos de experiencia en la productividad de los puestos de trabajo relacionados con los proyectos de desarrollo y mantenimiento de software.

5.2.2. Caso de estudio 2

5.2.2.1. Descripción de organización y proceso de trabajo

La organización participante es una organización europea cuya actividad se centra en la consultoría, integración de sistemas y soluciones, *outsourcing* de aplicaciones y servicios TIC; siendo los servicios TIC su principal negocio, con un volumen de negocio total, en 2012, de más de 1.200 millones de Euros. Esta organización cuenta con unos 14.000 empleados en diferentes países.

Dentro de esta organización, en el área de *outsourcing* de aplicaciones, se encuentra el proyecto dentro del cual se enmarca la participación. El proyecto es tiene como objetivo el mantenimiento evolutivo y correctivo de un producto *Front End* previamente desarrollado para una administración pública. El proyecto se desarrolla bajo un acuerdo de nivel de servicio con una duración de contrato de duración determinada y un presupuesto anual de 4M €; el producto ya estaba en producción al inicio del contrato. El proyecto se lleva a cabo en un entorno de *nearshore* (500-1000 Km) entre distintos países, compartiendo el idioma entre el cliente y casi toda la plantilla de la organización participante.

El producto está desarrollado en *JAVA* con una capa de negocio en *COBOL* y una base de datos *DB2*. Cuenta con unos 30.000 usuarios y entorno a 10 millones de

personas *gestionadas*. La capa de negocio COBOL es desarrollada por otra organización, y la base de datos está mantenida por el cliente. Además, el cliente es el encargado de la integración final. Anualmente se desarrollan tres o cuatro versiones con las mejoras y correcciones asignadas a dicha versión. Se destinan en torno a tres semanas a desarrollo, dos a test, y uno a control y validación. En producción se encuentran tres o cuatro versiones en paralelo.

5.2.2.2. Puestos de trabajo, variables y factores a analizar

En el proyecto participan en torno a 20 personas en tres ubicaciones, por lo que se produce un desarrollo distribuido: dos en el país del cliente, 11 en un centro de trabajo (Centro de Trabajo A) y siete en otro (Centro de Trabajo B) de la organización participante, separados entre sí unos 500 Km. Estas 20 personas se distribuyen de la siguiente forma:

- Director de Proyectos: 1 en Centro de Trabajo A.
- Experto Funcional: 1 en país cliente.
- Experto Técnico: 1 en país cliente.
- Jefe de Proyecto: 1 en Centro de Trabajo A.
- Adjuntos Jefe de Proyecto: 2 en Centro de Trabajo A.
- Responsables de Versión: 3 en Centro de Trabajo A.
- Colaboradores: 7 en Centro de Trabajo B y 4 en Centro de Trabajo A. Estos colaboradores, en función de la experiencia (de menor a mayor) son identificados como refuerzo, nominal, y experto; siendo los expertos el equipo base del proyecto.

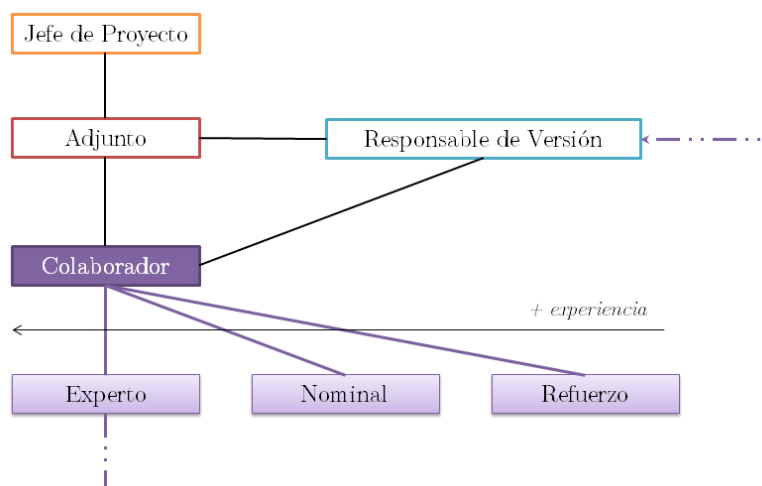


Ilustración 20. Organigrama de puestos en el Caso 2

Desde el punto de vista de la definición de los puestos:

- Colaborador: Este puesto se asemeja a un puesto de Ingeniero de Software, el cuál puede encontrarse como *Software developer*¹⁷ en O*NET. Es un puesto de trabajo que desempeña múltiples tareas, y al que se le asignan las tareas en función de las características propias de cada persona por parte del responsable de versión.
- Jefe de Proyecto: Este puesto lleva a cabo la gestión integral del proyecto, mediante la supervisión del mismo. El objetivo es que sean autónomos, como si se tratara de un empresario que tiene que llevar a cabo un proyecto.
- Responsable de Versión: Este puesto realiza las tareas de gestión de la versión que tiene asignada durante toda la vida de la misma. Se trata del Jefe de Proyecto del día a día de esa versión en concreto. Este puesto de trabajo asigna tareas a los colaboradores que participan en la versión de la cual es responsable intentando que se produzca un balanceo de dificultad para que no todas las tareas difíciles o fáciles recaigan sobre un núcleo de

¹⁷ <http://www.onetonline.org/link/summary/15-1132.00>

personas, es decir, permitiendo que todos aprendan, y teniendo en cuenta que los test los realice una persona distinta a la que desarrolla la necesidad a probar. Esta asignación la realiza de forma oral en el caso de trabajadores en el mismo centro de trabajo, o por medio de telecomunicación (llamada y/o email) para trabajadores en distinto centro.

Para más información sobre el proceso de trabajo seguido por estos puestos de trabajo consultar el Anexo G - Diagramas de procesos de proyectos de desarrollo software en las organizaciones participantes (Caso 2).

El puesto de trabajo objeto de estudio en este caso es el de "Colaborador", y como variables de estudio se plantearon las siguientes:

	Nombre	Periodicidad	Escala
Entradas	Esfuerzo	Cada tarea	Entero (Horas)
	Experiencia - Tecnología	1 vez, al principio	Entero (Trimestres)
	Antigüedad	1 vez, al principio	Entero (Trimestres)
	Experiencia - Funcional	1 vez, al principio	Entero
	Motivación	1 vez, al principio de cada versión	Indicador
	Formación	1 vez, al principio	Ordinal
	Estimación	Cada tarea	Entero (Horas)
Salidas	Tareas finalizadas	Cada tarea	Entero
	Dificultad de tarea	Cada tarea	Likert-5 (Ninguna, Poca, Mediana, Bastante, Máxima)
	Calidad de la realización	Versión, al final	Entero
	Calidad de las especificaciones	Versión, al final	Entero
	Satisfacción	1 vez, al final	Indicador
Factores	Ubicación	1 vez, al principio	Categórico
	Retrasos	Cada tarea	Likert-4 (Nulo, Leve, Notable, Severo)
	Cambio de tarea	Cada tarea	Entero
	Interrupciones	Cada tarea	Entero

	Nombre	Periodicidad	Escala
	Puntos de entrada	Cada tarea	Entero

Tabla 35. Variables bajo estudio en el Caso 2

Como puede apreciarse, las variables giran en torno a la tarea como unidad de trabajo. En concreto, las variables utilizadas se definen de la siguiente manera:

- Esfuerzo: horas necesarias (e imputadas en el sistema de gestión de trabajo) para llevar a cabo la tarea.
- Experiencia - Tecnología: número de meses que el participante tiene de experiencia con la tecnología de desarrollo empleada (*Java*).
- Antigüedad: número de meses que el participante tiene de antigüedad en el proyecto.
- Experiencia - Funcional: número de versiones en las que el participante ha desempeñado tareas de análisis funcional.
- Motivación: motivación del participante orientada a la tarea y a la versión (proyecto). Para medir la motivación, se ha empleado el formulario creado por Warr, Cook y Wall (2004) que mide la motivación intrínseca a partir de la definición que los creadores del mismo utilizan: grado con el que una persona quiere desempeñar bien en su trabajo para alcanzar la satisfacción intrínseca. Esta definición parte de las dos facetas de involucramiento en el trabajo definidas por Lodhal y Kejner (1965). Ver Anexo H - Cuestionarios utilizados en los casos de estudio para más información.
- Formación: grado máximo de titulación oficial alcanzado (Formación Profesional, Carrea técnica, Carrera superior, Postgrado).
- Estimación: estimación de la carga de trabajo necesaria para cada tarea. Dado que no todas las tareas son estimadas, aquellas que no tengan estimación se le asigna como estimación la carga de trabajo incurrida, por lo que para dichas no hay desvío de estimación.
- Tareas finalizadas: tareas asignadas y terminadas por un colaborador.

- Dificultad de tarea: valoración por el colaborador de la dificultad de la tarea realizada.
- Calidad de la realización: valoración interna de la calidad del código fuente desarrollado a nivel de versión. Esta variable no se analiza inicialmente dentro del modelo DEA ya que sería constante para todos los participantes al ser una variable de versión y sólo disponer de datos de una versión; así pues, quedaría como variable de estudio por si se analizaran distintas versiones y/o para comparar productividad entre versiones.
- Calidad de las especificaciones: número de aclaraciones adicionales necesarias para desarrollar una funcionalidad o cambio solicitado en la especificación de requisitos.
 - Satisfacción: grado de satisfacción del colaborador con el trabajo realizado con respecto a la naturaleza del trabajo, las condiciones de trabajo, y la comunicación (Spector, 1985). Ver Anexo H - Cuestionarios utilizados en los casos de estudio para más información.
- Ubicación: localización donde el colaborador desempeña su trabajo: Centro de trabajo A (CT-A) o B (CT-B).
- Retrasos: valoración del colaborador del tiempo perdido/retraso provocado por el “entorno” para desempeñar su tarea.
- Cambio de tarea: número de veces que el colaborador cambia de tarea antes de finalizarla.
- Interrupciones: número de veces que el colaborador tiene que interrumpir la tarea por motivos ajenos.
- Puntos de entrada: número de cambios en las especificaciones de requisitos desde su toma inicial.

5.2.2.3. Descripción demográfica de participantes

Los participantes son colaboradores de una misma versión y pertenecen a los dos centros de trabajo de España. La descripción demográfica, de los participantes que inicialmente iban a participar, es la siguiente:

ID	Edad (<i>a 31/12/2012</i>)	Sexo	Estudios
01	27	Mujer	Carrera Superior
02	36	Hombre	Carrera Superior
03	36	Hombre	Carrera Superior
04	42	Hombre	Posgrado
05	28	Hombre	Carrera Superior
06	33	Hombre	Carrera Técnica
07	34	Hombre	Carrera Superior
08	28	Mujer	Carrera Superior
09	32	Hombre	Posgrado
10	27	Hombre	Carrera Técnica
11	29	Mujer	Carrera Superior
12	29	Mujer	Carrera Superior
13	28	Mujer	Posgrado
14	28	Mujer	<i>No indicado</i>
15	27	Hombre	Carrera Superior
16	39	Hombre	Carrera Superior
17	38	Hombre	Formación Profesional

Tabla 36. Descripción demográfica de los participantes (iniciales) en el Caso 2

Por motivos varios, dado que no todos los participantes finalizaron la participación, la muestra final de participantes es la siguiente:

ID	Edad (<i>a 31/12/2012</i>)	Sexo	Estudios	Centro de Trabajo
01	27	Mujer	Carrera Superior	CT-B
02	36	Hombre	Carrera Superior	CT-B
03	36	Hombre	Carrera Superior	CT-A
05	28	Hombre	Carrera Superior	CT-A

ID	Edad (<i>a</i> 31/12/2012)	Sexo	Estudios	Centro de Trabajo
08	28	Mujer	Carrera Superior	CT-A
10	27	Hombre	Carrera Técnica	CT-A
11	29	Mujer	Carrera Superior	CT-A
12	29	Mujer	Carrera Superior	CT-B
13	28	Mujer	Posgrado	CT-B
16	39	Hombre	Carrera Superior	CT-B
17	38	Hombre	Formación Profesional	CT-A

Tabla 37. Descripción demográfica de los participantes (finales) en el Caso 2

La muestra final consta de 11 participantes, de los cuales seis hombres (edad media 34, *de* 5,177), cinco mujeres (edad media 28,2, *de* 0,84); seis pertenecen al Centro de Trabajo A, y cinco al B. Con respecto a la formación: ocho disponen de Carrera Superior (72,73%), uno de Posgrado, otro de Carrera Técnica, y otro Formación Profesional (9,09%).

5.2.2.4. Preparación de datos

Pese a que los datos han sido recogidos de forma bisemanal, el tratamiento se ha realizado una vez que todos los datos se recibieron. No obstante, tras la primera entrega se comprobado la disponibilidad de las variables objeto de medición así como su correcto formato. Es necesario comentar que algunas variables han tenido que ser obtenidas de forma manual, ya que los sistemas de gestión de trabajo no han permitido una automatización personalizada de recogida de información.

A partir de los datos obtenidos, se han comprobado las correlaciones entre las variables con un doble objetivo: (1) reducir variables del mismo tipo que correlacionasen entre sí, y (2) realizar el análisis de isotonicidad, tal y como requiere el uso de DEA. En concreto se obtuvieron varias correlaciones; algunas de las variables sólo se relacionan con una o dos variables, mientras que otras correlacionan con un conjunto mayor de variables. En las siguientes tablas se incluyen las correlaciones

estadísticamente significativas encontradas (se marca con * las correlaciones cuya *sig.* es menor que 0,005 y con ** las que son menores a ,001):

	Desvío Acumulado	Cambios de Tarea	Experiencia Funcional	Jornadas	Jornadas Estimadas	Calidad de las Especificaciones	Antigüedad	Interrupciones	Interrupciones por Tarea
Experiencia tecnológica	,941* (.034)	,599* (.025)							
Antigüedad			,993* (.018)	-,629* (.038)	-,627* (.039)	-,574* (.023)			
Experiencia Funcional						-,632 (.037)	,933* (.018)	,933* (.018)	,898* (.039)
Tareas									
Jornadas				,618* (.043)	,648* (.031)				,824** (.002)
Jornadas Estimadas		,775** (.009)		,984** (.000)	,984** (.000)				
		,957* (.039)				,574* (.023)			

Tabla 38. Correlaciones entre variables en el caso de estudio 2 (1/2)

	Satisfacción (Naturaleza del Trabajo)	Satisfacción (Condiciones de Trabajo)	Satisfacción (Comunicación)	Satisfacción Total	Puntos de Entrada	Calidad de la Realización	Retrasos (Nulo)	Retrasos (Notable)	Dificultad (Poca)
Motivación	,795** (.003)	,740** (.000)	,782** (.004)	,831** (.001)		,612* (.033)			
Satisfacción (Naturaleza del Trabajo)		,651* (.03)	,902** (.000)	,928** (.000)		,629* (.038)			
Satisfacción (Condiciones de Trabajo)			,854** (.001)	,882** (.000)	,790** (.004)	,705** (.003)			
Satisfacción (Comunicación)				,981** (.000)	,740** (.000)	,811** (.002)			
Satisfacción Total					,711* (.014)	,788** (.007)			,865** (.003)
Tareas					,731* (.011)	,760** (.007)			
Calidad de las Especificaciones					,841** (.001)	,798** (.003)	,831** (.002)		
Puntos de Entrada						,970** (.000)	,727* (.011)	1,000** (.000)	
Calidad de la Realización							,729* (.011)	,986* (.014)	
Retrasos (Nulo)									,849** (.004)

Tabla 39. Correlaciones entre variables en el caso de estudio 2 (2/2)

Tras analizar las correlaciones iniciales, y con el objetivo de reducir la cantidad de variables a incluir en el modelo DEA y poder conocer la isotonicidad existente, tal y como se ha descrito en 4.2 Data Envelopment Analysis (DEA), se han realizado las siguientes operaciones:

- Jornadas: Se ha eliminado de la lista de variables ya que correlaciona de forma estadísticamente significativa con otra variable del mismo tipo (entrada) "Jornadas Estimadas", la cual tiene un significado parecido y

tienen correlación estadísticamente significativa con "Cambios de Tarea". Además, "Jornadas Estimadas" correlaciona de forma estadísticamente significativa con "Calidad de las Especificaciones" lo que no se produce con "Jornadas" de forma estadísticamente significativa, no obstante la significancia roza el límite ($,053 > ,005$) por lo que también podrían aceptarse como correlacionados si dicho grado fuera ligeramente más amplio.

- Calidad de las Especificaciones, Calidad de la Realización, y Puntos de Entrada:
 - Dado que son medidas tomadas a nivel de tarea se ha dividido la cantidad de tareas realizadas por la DMU entre el doble del total de dicho valor, al que se le ha sumado 0,75 para evitar valores nulos los cuales producirían una división por 0.

-
- Se ha optado por el doble del valor para primar la puntuación de estas variables ante la cantidad de tareas, es decir, es más importante el valor que la cantidad de tareas realizadas.
 - Se ha optado por 0,75 como constante para anular los valores nulos ya que es un valor que no aumenta de forma considerable los casos nulos. Si se hubiera optado por sumar 0,1 las divisiones hubieran originado grandes diferencias entre valores no nulos y valores nulos.

Tras estas transformaciones, se han seleccionado las siguientes variables, que cumplen el requisito de Isotonicidad:

	Calidad de las especificaciones	Satisfacción	Calidad de la Realización
Experiencia Funcional	-,632 (,037)		
Jornadas Estimadas	,674* (,023)		
Motivación		,834** (,001)	,642* (,033)
Puntos de Entrada	,841** (,001)	,711* (,014)	,970** (,000)

	Calidad de las especificaciones	Satisfacción	Calidad de la Realización
Calidad de las Especificaciones			,798** (,003)
Satisfacción			,788** (,007)

Tabla 40. Correlaciones (a partir de valores sin transformar)¹⁸

Estas variables tienen los siguientes valores para cada DMU (participante):

ID	Experiencia Funcional	Motivación	Satisfacción	Jornadas Estimadas	Calidad de las Especificaciones	Puntos de Entrada	Calidad de la Realización
01	1	32	63	19,25	0,7451	1,6522	1,4074
02	1	30	59	18,25	0,6957	5,3333	2,2857
03	4	29	50	1,75	2,6667	2,6667	2,6667
05	5	30	61	6,50	1,3333	1,3333	1,3333
08	1	27	47	19,50	0,5143	2,5714	6,0000
10	1	27	45	1,50	0,6667	0,6667	0,6667
11	3	26	46	2,25	2,6667	2,6667	2,6667
12	1	29	47	22,25	0,5263	3,3333	3,3333
13	1	35	72	13,75	0,5128	0,8696	0,7407
16	1	30	55	15,75	2,0000	7,3333	3,1429
17	5	34	55	14,75	4,6667	4,6667	4,6667

Tabla 41. Valores para las DMUs (Caso 2)

5.2.2.5. Generación y análisis de resultados

Con las variables disponibles tras la comprobación de requisitos y transformaciones, se plantean dos modelos de productividad, cuyo nombre ha sido inventado para denominarlos de aquí en adelante:

¹⁸ * = Sig < ,05 | ** = Sig < ,01

- Productividad de Desarrollo de Software. Para este análisis se utilizan las siguientes variables:
 - Entradas:
 - Motivación.
 - Salidas:
 - Satisfacción.
 - Calidad de la Realización.
- Productividad de Análisis de Necesidades. Para este análisis se utilizan las siguientes variables:
 - Entradas:
 - Experiencia Funcional.
 - Puntos de Entrada.
 - Salidas:
 - Calidad de las Especificaciones.

5.2.2.5.1. Productividad de Desarrollo de Software

Se ha empleado el modelo BCC orientado a salidas con eficiencia tecnológica y retornos variables a escala; y los resultados han sido los siguientes:

ID	Eficiencia	DMUs eficientes de referencia (valor)		
		1	2	3
01	0,972	05 (0,47)	08 (0,081)	13 (0,449)
02	0,994	05 (0,641)	08 (0,224)	13 (0,134)
03	0,890	05 (0,627)	08 (0,358)	13 (0,015)
05*	1			
08*	1			
10	0,905	05 (0,25)	11 (0,75)	
11*	1			
12	0,860	05 (0,301)	08 (0,562)	13 (0,137)
13*	1			

ID	Eficiencia	DMUs eficientes de referencia (valor)		
		1	2	3
16	0,954	05 (0,272)	08 (0,455)	13 (0,273)
17*	1			

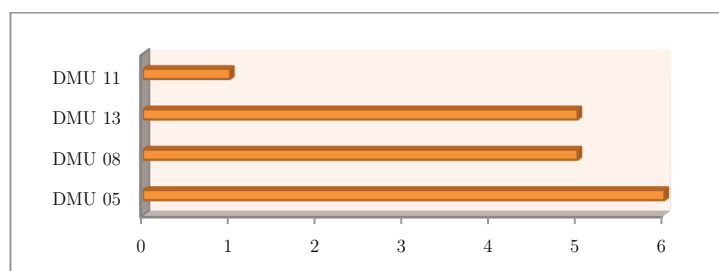
Tabla 42. Productividad de Desarrollo de Software (BCC)¹⁹

Ilustración 21. Número de DMUs ineficientes que consideran a otra DMU como modelo de referencia

Se han analizado las holguras para el análisis realizado y se han encontrado los siguientes valores:

- Calidad de la Realización:
 - DMU 10: 1,596.
- Motivación: Sin holguras.
- Satisfacción: Sin holguras.

Estas holguras, al ser casi inexistentes, nos indican que los resultados obtenidos mediante el modelo radial BCC son representativos. Por otro lado, pese a que es posible obtener los puntos de mejora utilizando las proyecciones de cada DMU sobre la frontera de eficiencia y ajustarlo con las holguras, se ha decidido emplear un segundo modelo DEA. En concreto, se ha empleado el modelo SBM con eficiencia mixta y retornos a escala variables. Orientando el modelo SBM a salidas, se han obtenido los siguientes resultados:

¹⁹ * DMUs eficientes

ID	Eficiencia	DMUs eficientes de referencia (valor)		
		1	2	3
01	0,721	05 (0,081)	08 (0,324)	13 (0,595)
02	0,956	05 (0,568)	08 (0,27)	13 (0,162)
03	0,657	08 (0,721)	13 (0,045)	17 (0,234)
05*	1			
08*	1			
10	0,199	08 (1)		
11*	1			
12	0,714	08 (1)		
13*	1			
16	0,839	08 (0,613)	13 (0,288)	17 (0,099)
17*	1			

Tabla 43. Productividad de Desarrollo de Software (SBM)²⁰

Con este método las DMUs eficientes son las mismas que las obtenidas con el modelo BCC pero cambian las DMUs de referencia así como los pesos para las DMUs ineficientes. También se observa una fuerte caída del indicador para la DMU 10, que pasa de 0,905 a 0,199. Este cambio de modelo convierte a esta DMU en muy improductiva. En menor medida las DMUs 01 y 12 descienden también.

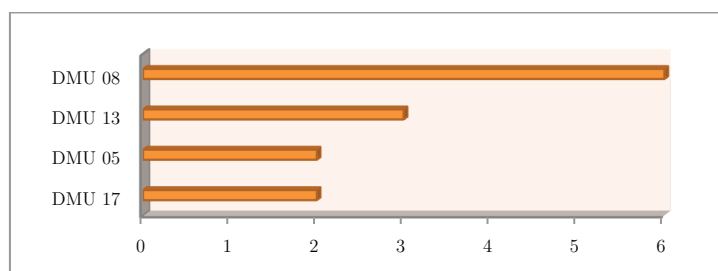


Ilustración 22. Número de DMUs ineficientes que consideran a otra DMU como modelo de referencia (SBM)

²⁰ * DMUs eficientes

Las holguras obtenidas para las DMUs ineficientes son las siguientes (entre paréntesis el porcentaje de mejora relativo necesario para que la DMU sea productiva):

DMU	Motivación	Satisfacción	Calidad de la Realización
01	-	-	1,088 (77%)
02	-	-	0,212 (9%)
03	-	-	2,784 (104%)
10	-	2 (4%)	5,333 (800%)
12	2 (7%)	-	2,667 (80%)
16	-	-	1,209 (38%)

Tabla 44. Holguras mediante SBM en Productividad de Desarrollo

Las holguras identificadas por este modelo incluyen la detectada mediante BCC (DMU 10, Calidad de la Realización) y añaden otras nuevas, principalmente en la variable Calidad de la Realización, por lo que el conjunto de holguras es mayor, tal y como era de esperar. Entre las DMUs ineficientes, la mejora pasa principalmente por un aumento de la Calidad de la Realización (184%), y casi de forma despreciable por una reducción de la Motivación (en un 1% de media) y un aumento de la Satisfacción (<1%). Dado que la mayor cantidad de holguras se detecta en la variable Calidad de la Realización y que el peso de la mejora es indudablemente mayor que para las otras variables (184% *vs* 1% y <1%) tiene sentido plantear esta variable como elemento prioritario a mejorar.

5.2.2.5.2. Productividad de Análisis de Necesidades

Los resultados para este modelo han sido los siguientes:

ID	Eficiencia	DMUs eficientes de referencia (valor)	
		1	2
01	0,862	10 (0,852)	16 (0,148)
02	0,435	10 (0,3)	16 (0,7)
03	1	10 (0,5)	17 (0,5)

ID	Eficiencia	DMUs eficientes de referencia (valor)	
		1	2
05	1	10 (0,833)	17 (0,167)
08	0,491	10 (0,714)	16 (0,286)
10*	1		
11*	1	10 (0,5)	17 (0,5)
12	0,438	10 (0,6)	16 (0,4)
13	0,725	10 (0,97)	16 (0,03)
16*	1		
17*	1		

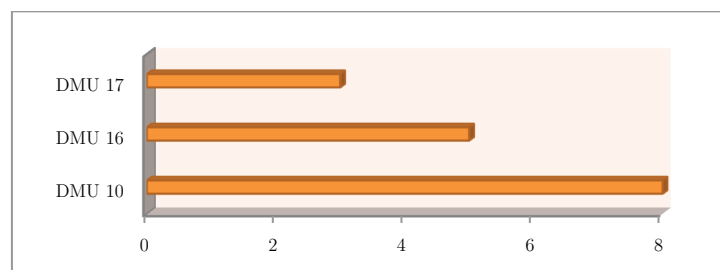
Tabla 45. Productividad de Análisis de Necesidades (BCC)²¹

Ilustración 23. Número de trabajadores ineficientes que consideran a otro como modelo de referencia

Se han analizado las holguras para el análisis realizado y se han encontrado los siguientes valores:

- Experiencia Funcional:
 - DMU 03: 1.
 - DMU 05: 3,334.
- Puntos de Entrada: Sin holguras.
- Calidad de las Especificaciones: Sin holguras.

²¹ * DMUs eficientes

Estas holguras, al ser casi inexistentes, nos indican que los resultados obtenidos mediante el modelo radial BCC son representativos. Por otro lado, pese a que es posible obtener los puntos de mejora utilizando las proyecciones de cada DMU sobre la frontera de eficiencia y ajustarlo con las holguras, se ha decidido emplear un segundo modelo DEA. En concreto, se ha empleado el modelo SBM con eficiencia mixta y retornos a escala variables. Orientando el modelo SBM a entradas, se han obtenido los siguientes resultados:

ID	Eficiencia	DMUs eficientes de referencia (valor)	
		1	2
01	0,82	10 (0,941)	16 (0,059)
02	0,576	10 (0,978)	16 (0,022)
03	0,875	10 (0,5)	17 (0,5)
05	0,667	10 (0,834)	17 (0,167)
08	0,63	10 (1)	
10*	1		
11*	1	10 (0,5)	17 (0,5)
12	0,6	10 (1)	
13	0,883	10 (1)	
16*	1		
17*	1		

Tabla 46. Productividad de Desarrollo de Análisis de Necesidades (SBM)²²

En esta medida de productividad, las DMUs que tienen una medida de productividad de valor unitario pero no son productivas, tienen un menor resultado (DMU 03: 0,875, y DMU 05: 0,667). Con respecto a las demás DMUs los cambios son poco significativos ya que se mantienen las DMUs eficientes y los valores de las ineficientes son muy parecidos observándose un ligero ascenso de los resultados.

²² * DMUs eficientes

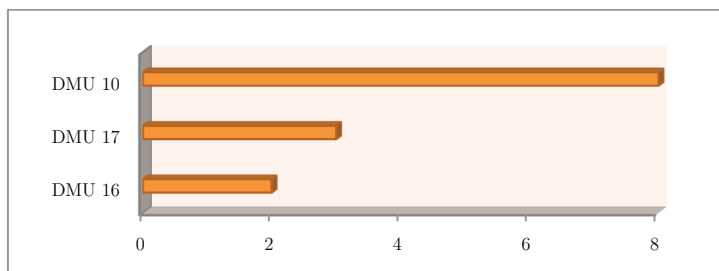


Ilustración 24. Número de DMUs ineficientes que consideran a otra DMU como modelo de referencia (SBM)

Las holguras que se han obtenido para las DMUs ineficientes son las siguientes (entre paréntesis el porcentaje de mejora relativo necesario para que la DMU sea productiva):

DMU	Experiencia Funcional	Puntos de Entrada	Calidad de las Especificaciones
01	-	0,595 (36%)	-
02	-	4,521 (85%)	-
03	1 (25%)	-	-
05	3,334 (67%)	-	-
08	-	1,904 (74%)	0,153 (30%)
12	-	2,666 (80%)	0,141 (27%)
13	-	0,203 (23%)	0,154 (30%)

Tabla 47. Holguras mediante SBM en Productividad de Análisis de Necesidades

Las holguras identificadas por este modelo incluyen las obtenidas con BCC y se identifican otras en las otras dos variables analizadas, por lo que el conjunto de holguras detectadas con SBM es mayor, tal y como era de esperar. Entre las DMUs ineficientes, la mejora pasa por una reducción de la Experiencia Funcional (en un 13% de media), reducción de los Puntos de Entrada (43%) y el aumento de la Calidad de las Especificaciones (12%). Dado que la mayor cantidad de holguras se detecta en la variable Puntos de Entrada y que el peso de la mejora es mayor que para las otras variables (43% *vs* 12% y 13%) tiene sentido plantear esta variable como elemento prioritario a mejorar.

5.2.2.6. Conclusiones parciales

- Se han realizado dos análisis de productividad para un mismo grupo de trabajadores dentro de un mismo puesto utilizando el modelo BCC. Posteriormente, se ha aplicado el modelo SBM para identificar las holguras con el objetivo de encontrar los puntos de mejora. Este segundo paso también podría haberse realizado con los datos del modelo BCC pero ha sido de utilidad para comprobar que las DMUs productivas casi se mantienen en su totalidad con ambos modelos para los casos analizados.
- En cada uno de los análisis se ha identificado un conjunto de trabajadores referencia para cada análisis, es decir, DMUs eficientes que son identificadas por otras DMUs improductivas como referencia de eficiencia. En concreto:
 - Se han encontrado las siguientes DMUs referentes (ordenadas de más referenciada a menos):
 - Según el modelo BCC:
 - Productividad de Desarrollo: 05 (6 veces), 08 (5), 13 (5), y 11 (1).
 - Productividad de Análisis de Necesidades: 10 (8 veces), 16 (5), y 17 (3).
 - Según el modelo SBM:
 - Productividad de Desarrollo: 05 (6 veces), 08 (6), 13 (3), 05 (2), y 17 (2). Aparece referenciado las DMUs 05 y 17, y desaparece la DMU 11. Las DMUs 05 y 13 se mantienen como DMUs de referencia con un número de referencias parecido.
 - Productividad de Análisis de Necesidades: 10 (8 veces), 16 (3), y 17 (2). Las mismas DMUs y casi con el mismo número de referencias que con el modelo BCC.
 - Se han encontrado dos DMUs con valor de eficiencia 1 que no son identificadas como eficientes para la productividad de Análisis de Necesidades, según el modelo BCC. En concreto se trata de las DMUs 03 y 05, las cuales tienen una holgura de Experiencia Funcional de 1 y 3,334 respectivamente; es decir, para que estas DMUs fueran consideradas eficientes habrían que reducir su Experiencia Funcional en el valor indicado por la holgura.
- Se han detectado los puntos de mejora de las DMUs ineficientes utilizando el modelo SBM. Desde el punto de vista de la Productividad de

Desarrollo, la mejora pasaría por el **aumento de la calidad de la realización**, esto es, por la reducción de la cantidad de fallos encontrados en la fase de desarrollo. Y desde el punto de vista de la Productividad de Análisis de Necesidades, la mejora pasaría por la **reducción de la cantidad de Puntos de Entrada**, esto es, por la reducción del número de cambios asociados a los requisitos de cada tarea, o lo que es lo mismo, por el aumento de la calidad de los requisitos previamente al desarrollo de la funcionalidad y/o cambios requeridos.

- Algunas **correlaciones** de la muestra pueden ser destacas:
 - La **experiencia funcional correlaciona de forma positiva con las interrupciones** (0,898; *Sig* ,039). Esto puede ser debido a que las personas con mayor experiencia funcional son más consultadas e interrumpidas y/o tienen más cantidad de tareas que atender.
 - La **motivación correlaciona de forma positiva con la satisfacción** (0,834; *Sig* ,001); **y con los subitems de satisfacción utilizados**.
- Finalmente, comentar que a pesar de la cantidad de variables recogidas tras comprobar la validez de los datos para aplicar DEA, las variables han quedado reducidas a un número muy inferior. De este modo, **en futuros estudios se debería partir de una base de variables mucho mayor o partir de una base donde se haya probado anteriormente que las variables de entrada y las variables de salida correlacionan** (positivamente).

5.2.3. Caso de estudio 3

5.2.3.1. Descripción de organización y proceso de trabajo

La organización participante es una organización española cuya actividad se centra en la distribución de contenidos digitales editoriales de un área temática; siendo los servicios TIC una de las áreas facilitadoras de negocio dada la transformación que ha sufrido la producción de contenidos en las últimas décadas, en las que se ha pasado de distribución en papel a distribución multiplataforma en tiempo real y bajo demanda. La organización de este caso es una marca dentro de una organización internacional de mayor tamaño (en la que también se encuentra la organización del caso de estudio 4). Esta organización internacional cuenta con un volumen de negocio, en 2012, de más de

3.400 millones de Euros, con unos 19.000 empleados en diferentes países, y está considerada como una organización con buenas prácticas de trabajo (*Great Places To Work*[®]).

Dentro de la organización, el área de desarrollo y mantenimiento de software es el área en la que se encuentran los puestos de trabajo a analizar. El cliente de los proyectos es la propia organización (cliente interno), y el objetivo de los mismo es llevar a cabo un mantenimiento evolutivo y correctivo de los proyectos previamente desarrollados. Debido a la gran variedad de proyectos desarrollados (hasta 30 proyectos en paralelo), la principal actividad del área de desarrollo es el mantenimiento evolutivo y correctivo de los proyectos, quedando en un segundo plano las actividades relacionadas con nuevos desarrollos.

Los proyectos suelen ser desarrollados en *.NET* con accesos a bases de datos (*Oracle*) o bien en *PHP* y *MySQL*, excepcionalmente se emplea *SQL* para las bases de datos. En cuanto a la tipología de proyectos se puede decir que en su mayoría son proyectos de repositorios de información que utilizan entornos web como medio de interacción con los usuarios. La organización no sigue una metodología estandarizada aunque se basa en metodologías ágiles (p. ej., empleo de *sprints* para asignar tareas). Además dispone de un flujo de trabajo bien organizado y soportado en un software de gestión de proyectos. Añadir, que los proyectos no son siempre desarrollados y mantenidos por el mismo equipo y que la organización dispone de varios centros de trabajo ubicados en España.

5.2.3.2. Puestos de trabajo, variables y factores a analizar

En el área de desarrollo de software hay trabajando alrededor de 50 personas asignadas a los puestos de la siguiente manera:

- Jefe de Equipo/Proyecto: 4 personas.

- Analistas: 16 personas.
- Analistas Programadores: 20 personas.
- Programadores: 20 personas.

Los puestos de trabajo involucrados en el desarrollo y mantenimiento de proyectos software se definen de la siguiente manera:

- Jefe de Equipo: este puesto tiene asignada la gestión de tareas enfocadas al proyecto: asignación y control de tareas, solución de problemas en producción (servicios y herramientas)..., y es el responsable de la organización de personas a su cargo: planificación de vacaciones, días libres, problemas personales... El enfoque de gestión se centra en el aspecto técnico del proyecto, dejando fuera la labor de costes. Es un facilitador de problemas que puedan surgir en el día a día del proyecto. Además, es un punto de entrada para muchas tareas y puestos.
- Analista: este puesto lleva a cabo un análisis de tareas, las fragmenta, y asesora a analistas-programadores y programadores en su trabajo. Interactúa con el *Product Owner* (Responsable de Producto) para aclarar y definir con más detalle los requisitos. Además, colabora con el *Jefe de Equipo* para que el proyecto sea exitoso.
- Analista Programador: este puesto recoge los requisitos que han elaborado los *Analistas* (desde la funcionalidad requerida por los *Product Owner*), realiza un análisis más detallado y establece con los Analistas un reajuste de la estimación y desglose de los requisitos. Una vez se han realizado estas labores, el *Analista-Programador* participa en el desarrollo software del proyecto como tal y como un *Programador*.
- Programador: este puesto desarrolla el código fuente de proyectos relativo a las tareas asignadas. Además realiza mantenimientos correctivos y evolutivos de varios proyectos. También lleva a cabo consultas a base de datos.

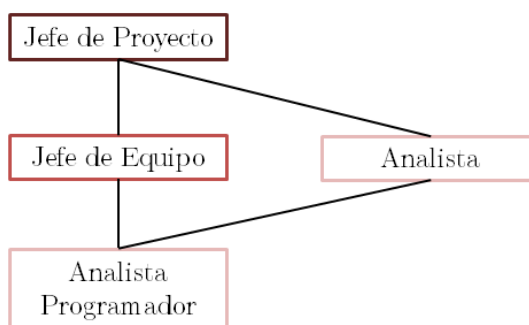


Ilustración 25. Organigrama de puestos en el Caso 3

Para más información sobre el proceso de trabajo seguido por estos puestos de trabajo consultar el Anexo G - Diagramas de procesos de proyectos de desarrollo software en las organizaciones participantes (Caso 3).

Inicialmente se plantean estas variables para los puestos de trabajo que finalmente serán objeto de estudio en este caso (Analista y Analista Programador):

	Nombre	Periodicidad	Escala
Entradas	Esfuerzo	Tarea	Entero (Horas)
	Experiencia en el puesto	1 vez	Entero (Meses)
	Experiencia en la tecnología	1 vez	Entero (Meses)
	Conocimiento del proyecto	1 vez	Likert
	Motivación	Proyecto	Indicador (Cuestionario)
	Formación	1 vez	Ordinal
	Hardware	Tarea	Likert
	Claridad de funcionalidad requerida	Tarea (<i>Análisis</i>)	Likert
	Claridad de requisitos	Tarea (<i>Codificación</i>)	Likert
	Estimación	Tarea	Entero
	Cambios de proyecto	Tarea	Entero
	Interrupciones	Tarea	Likert
Salidas	Tareas resueltas	Tarea	Entero
	Bugs resueltos	Tarea	Entero
	Dificultad de Tarea/Bug	Tarea	Likert
	Satisfacción del cliente	Proyecto	Indicador (Cuestionario)

	Nombre	Periodicidad	Escala
	Satisfacción personal (<i>Motivación extrínseca - Recompensa</i>)	Proyecto	Indicador (Cuestionario)
	Integración en el proyecto	Proyecto	Indicador (Cuestionario)
Factores	Ubicación	1 vez	Categorico
	Teletrabajo	1 vez	Booleano

Tabla 48. Variables bajo estudio en el Caso 3

Como puede apreciarse, las variables giran en torno a la tarea como unidad de trabajo, al igual que en el Caso 2. En este caso, las variables utilizadas se definen de la siguiente manera:

- Esfuerzo: número de horas que el participante imputa a cada tarea hasta el momento de ser dadas por resueltas.
- Experiencia en el puesto: número de meses que el participante lleva en el puesto actual.
- Experiencia en la tecnología: grado de experiencia en la tecnología utilizada en el proyecto del participante (contando también con la experiencia anterior a la actual organización).
- Conocimiento del proyecto: grado de conocimiento del proyecto del participante.
- Motivación: motivación del participante orientada a la tarea y la versión.
- Formación: grado máximo de titulación oficial alcanzado por el participante (Formación Profesional, Carrea técnica, Carrera superior, Postgrado).
- Hardware: valoración subjetiva que el participante tiene de la eficiencia del hardware para realizar cada tarea asignada.
- Claridad de funcionalidad requerida: valoración subjetiva que el participante tiene de la claridad de la funcionalidad requerida en la tarea asignada. Sólo se mide en tareas de análisis.

- Claridad de requisitos: valoración subjetiva que el participante tiene de la claridad de los requisitos relativos a la tarea asignada. Sólo se mide en tareas de programación.
- Estimación: estimación del esfuerzo necesario para llevar a cabo la tarea.
- Cambios de proyecto: número de cambios de proyecto que el participante tiene desde el inicio a la resolución de cada tarea.
- Interrupciones: valoración subjetiva que el participante tiene del grado de interrupción que sufre hasta cerrar cada tarea.
- Tareas resueltas: número de tareas en las que el participante participa en su ejecución y se dan por resueltas.
- Bugs resueltos: número de bugs en los que el participante ha participado en su resolución y han sido dados por resueltos.
- Dificultad de Tarea/Bug: valoración subjetiva que el participante tiene de la dificultad de cada tarea realizada.
- Satisfacción del cliente: valoración subjetiva que el cliente tiene de la satisfacción del cliente con respecto al proyecto.
- Satisfacción personal (Motivación extrínseca - Recompensa): valoración subjetiva que el participante tiene del grado de recompensa obtenido tras el periodo de medición.
- Integración en el proyecto: valoración subjetiva que el participante tiene del grado de integración del participante en el proyecto (*formar parte*).
- Ubicación: localización donde el participante desempeña su trabajo.
- Teletrabajo: booleano que indica si teletrabaja o no.

5.2.3.3. Descripción demográfica de participantes

La organización participante no ha continuado con la participación en la fase de validación al llegar a este punto, por lo que no ha sido posible continuar con este caso de estudio. Se ha intentado continuar con la participación en varias ocasiones pero no

se ha avanzado nada durante un periodo de seis meses por lo que se ha decidido cancelar la continuación de este caso.

5.2.3.4. Preparación de datos

No se han obtenido datos para este caso de estudio (ver apartado anterior).

5.2.3.5. Generación y análisis de resultados

En este caso de estudio no se han obtenido datos para su análisis por lo que se ha quedado como un caso teórico de medición de productividad para los puestos de trabajo de Analista y Analista Programador.

5.2.3.6. Conclusiones parciales

Para este caso no se han obtenido resultados de medición de productividad dado que la organización participante no ha facilitado la muestra de datos necesaria. No obstante, el caso ha permitido plantear un conjunto de variables de entrada y de salida, además de dos factores, que podrían emplearse para continuar con la medición en otro momento o en otra organización.

5.2.4. Caso de estudio 4

5.2.4.1. Descripción de organización y proceso de trabajo

La organización participante es una organización española cuya actividad se centra en el desarrollo de software de gestión para empresas. La organización de este caso es una marca dentro de una organización internacional de mayor tamaño desde hace algo más de una década (en la que también se encuentra la organización del caso de estudio 3). Esta organización internacional cuenta con un volumen de negocio, en 2012, de más de 3.400 millones de Euros, unos 19.000 empleados en diferentes países,

y está considerada como una organización con buenas prácticas de trabajo (Great Places To Work®).

En esta organización, los clientes finales de los proyectos son profesionales y organizaciones que utilizan algún producto de su gama de software de gestión. Las modificaciones y nuevos desarrollos se llevan a cabo bajo la petición de los *Product Owner*, que ejercen de clientes internos. La principal actividad del área de desarrollo es el mantenimiento evolutivo y correctivo de los productos ya desarrollados debido a externos vinculados con la gestión empresarial, de modo que quedan en un segundo plano las actividades relacionadas con nuevos desarrollos. El volumen de proyectos es reducido pero la actualización de los mismos es constante. Además de los proyectos de software, dispone de productos y servicios en formación y consultoría, herramientas de comunicación, y generación y distribución de contenidos.

Los proyectos suelen estar desarrollados en *.NET* con accesos a bases de datos (*Oracle*). En cuanto a la tipología de proyectos se puede decir que en su mayoría son proyectos de gestión de información que utilizan diversas plataformas para interactuar con el usuario. La organización se basa en metodologías ágiles (*SCRUM* y *MSF Agile*), y dispone del flujo de trabajo bien organizado y soportando en el software de gestión de proyectos empleado. Los proyectos no son siempre desarrollados y mantenidos por el mismo equipo y además la organización solo dispone de un centro de trabajo aunque los trabajadores pueden desempeñar su puesto en teletrabajo de hasta dos días por semana.

5.2.4.2. Puestos de trabajo, variables y factores a analizar

La organización participante dispone de varios roles definidos, que son los actores principales en el desarrollo y mantenimiento de proyectos software. Un trabajador puede desempeñar varios de estos roles por lo que el puesto de trabajo en sí mismo se asemeja al puesto de Ingeniero de Software, el cuál puede encontrarse como *Software*

*developer*²³ en O*NET. Un puesto similar se ha sido analizado en el Caso 2. A continuación se describen los principales roles para este puesto de trabajo:

- Gestor de desarrollo: este puesto crea la estructura del proyecto incluyendo las tareas necesarias, realiza el seguimiento de la ejecución del proyecto (en calidad, tiempo, y coste), y selecciona el equipo de trabajo necesario para llevar a cabo el proyecto.
- Analista funcional: este puesto apoya en la especificación de requisitos, lleva a cabo el diseño detallado a nivel funcional de los mismos siguiendo las guías de estilo, usabilidad y navegación establecidas. También lleva a cabo el diseño y ejecución de las pruebas funcionales que aseguren su correcta implementación. Además, presenta propuestas de mejoras funcionales, y colabora en la planificación y seguimiento de las peticiones, aportando las valoraciones de las tareas a realizar.
- Analista técnico: este puesto realiza el diseño técnico de forma que cumpla con la solución funcional diseñada y las guías de arquitectura correspondientes. Además, brinda un apoyo especializado al Analista funcional para garantizar la viabilidad de las soluciones funcionales planteadas. Además, lleva a cabo el control técnico de las soluciones implementadas, dando el soporte técnico necesario al programador para su realización, presenta propuestas de mejoras técnicas en los productos al Gestor de cliente, y colabora en la planificación y seguimiento de las peticiones, aportando las valoraciones de las tareas a realizar.
- Programador: este puesto implementa, en el periodo de tiempo planificado y siguiendo las guías de programación, buenas prácticas y arquitectura establecidas, las funcionalidades acordadas. El resultado debe integrarse con el código generado por el resto de programadores en el repositorio común

²³ <http://www.onetonline.org/link/summary/15-1132.00>

marcado en cada caso. Además, revisa el diseño técnico y construcción de la aplicación siguiendo las directrices fijadas, verifica la robustez de la construcción así como de la validación funcional de las funcionalidades construidas, y colabora en la planificación y seguimiento de las peticiones, aportando las valoraciones de las tareas a realizar.

- Analista de control de calidad: este puesto apoya al Analista de diseño en la definición de la solución, revisa y valida el plan de pruebas diseñado por los Analistas de diseño, selecciona las pruebas a ejecutar para validar el producto, apoya al Responsable del producto en la selección y marcado de las pruebas de aceptación, y colabora en la planificación y seguimiento de las peticiones, aportando las valoraciones de las tareas a realizar.

Para más información sobre el proceso de trabajo seguido por estos puestos de trabajo consultar el Anexo G - Diagramas de procesos de proyectos de desarrollo software en las organizaciones participantes (Caso 4).

Inicialmente se plantean las siguientes variables:

	Nombre	Periodicidad	Escala
Entradas	Desempeño	Tarea	Entero (Horas)
	Experiencia tecnología	1 vez	Entero (Meses)
	Conocimiento del proyecto	1 vez	Entero (Meses)
	Estimación	Tarea	Entero (Horas)
	Formación	1 vez	Ordinal
	Grado de avance de documentación	Tarea	Likert
	Experiencia funcional	1 vez	Entero (Meses)
	Claridad de requisitos	Tarea	Likert
	Grado de teletrabajo	1 vez	Entero
	Influencia hardware/conexiones en retraso	Tarea	Likert
	Percepción de interrupción	Tarea	Likert
	Interrupciones por dudas/preguntas	Tarea	Likert
Salidas	Tareas finalizadas	Tarea	Entero
	Bugs solucionados	Tarea	Entero

	Nombre	Periodicidad	Escala
	Dificultad de tarea/bug	Tarea	Likert
	Pruebas realizadas	Tarea	Entero
	Visión global del proyecto	Proyecto	Likert

Tabla 49. Variables bajo estudio en el Caso 4

Como puede apreciarse, las variables giran en torno a la tarea como unidad de trabajo, al igual que en los casos anteriores. En este caso, las variables utilizadas se definen de la siguiente manera:

- Desempeño: horas que son necesarias para llevar a cabo la tarea.
- Experiencia tecnología: número de meses que el participante tiene de experiencia con la tecnología de desarrollo empleada.
- Experiencia funcional: experiencia del participante en tareas de análisis funcional medido en meses.
- Conocimiento del proyecto: antigüedad en el proyecto.
- Estimación: número de horas estimadas para llevar a cabo la tarea.
- Formación: grado máximo de titulación oficial alcanzado (Formación Profesional, Carrea técnica, Carrera superior, Postgrado).
- Grado de avance de la documentación: escala del grado de avance de la documentación relativa a la tarea desempeñada en el momento de cerrarla.
- Claridad de requisitos: valoración del participante de la claridad de los requisitos de una tarea de análisis.
- Tareas finalizadas: tareas que se asignan y terminan.
- Bugs solucionados: número de bugs solucionados en la tarea.
- Dificultad de tarea: valoración por el participante de la dificultad de la tarea.
- Pruebas realizadas: número de pruebas realizadas.
- Visión global del proyecto: valoración del participante de cuanto de global es su visión sobre el proyecto al finalizar un ciclo de medición.

- Grado de teletrabajo: número de días que el participante puede desempeñar sus tareas en teletrabajo.
- Influencia hardware/conexiones en retraso: percepción del participante de la influencia que tiene el hardware y/o conexiones en el posible retraso de cierre de la tarea (desvío de la estimación).
- Percepción de interrupción: percepción del participante del grado de interrupciones que sufre durante la realización de la tarea.
- Interrupciones por dudas/preguntas: percepción del participante del grado de interrupciones por dudas y preguntas de otros compañeros durante la realización de la tarea.

5.2.4.3. Descripción demográfica de participantes

Los participantes son programadores del mismo equipo de trabajo realizando tareas para el mismo proyecto/producto. La descripción demográfica, de los participantes que participan es la siguiente:

ID	Edad (<i>a 31/12/2012</i>)	Sexo	Estudios
01	22	Mujer	Carrera Técnica
02	35	Hombre	Carrera Superior
03	35	Hombre	Formación Profesional
04	37	Hombre	Formación Profesional

Tabla 50. Descripción demográfica de los participantes en el Caso 4

La muestra queda de la siguiente manera: 4 participantes, de los cuales tres hombres (edad media 35,667 *de* 1,155), una mujer (edad 22). Con respecto a la formación: uno dispone de Carrera Superior (25%), uno de Carrera Técnica (25%), y dos de Formación Profesional (50%).

5.2.4.4. Preparación de datos

Los datos han sido recogidos en una sola fase, por lo que el tratamiento se ha realizado tras recibir los datos. De forma preliminar, se ha eliminado la variable de número de bugs solucionados ya que ningún participante ha indicado valores en ella. También se ha eliminado la variable claridad de requisitos ya que sólo un participante ha indicado valores en ella. Y, se ha creado una variable nueva a partir de otras dos existentes: *ahorro de planificación*, resultante de dividir el total de horas estimadas entre el total de horas imputadas para todas las tareas asignadas a la DMU.

Además de las anteriores modificaciones, las variables relacionadas con tarea para unificar valores en función de la cuantía de tareas realizadas se han modificado. De este modo, el valor total de las siguientes variables ha sido dividido entre la cantidad de tareas realizadas por cada DMU:

- Grado de avance de documentación.
- Dificultad de tarea.
- Pruebas realizadas.
- Influencia de hardware/conexiones en retraso.
- Percepción de interrupción.
- Interrupciones por dudas/preguntas.

A partir de los datos, se han comprobado las correlaciones entre las variables con un doble objetivo: reducir variables del mismo tipo que correlacionasen y realizar el análisis de isotonicidad, tal y como se requiere para aplicar DEA. En concreto se han obtenido varias correlaciones; algunas de las variables sólo se relacionan con una o dos variables, mientras que otras correlacionan con un conjunto mayor de variables. En las siguientes tablas se incluyen las correlaciones estadísticamente significativas (se marca con * las correlaciones cuya *sig.* es menor que 0,005 y con ** las que son menores a ,001) encontradas:

	Conocimiento del proyecto	Estimación	Preguntas/Tarea	Interrupciones por dudas y preguntas
Grado de teletrabajo	,980* (.02)			
Experiencia tecnológica		,972* (.028)		
Desempeño		,996** (.004)		
Dificultad/Tarea			-,991 (.009)	
Percepción de Interrupción				,996** (.004)

Tabla 51. Correlaciones entre variables Caso 4

Tras analizar las correlaciones, se concluye que no es posible plantear un modelo, dado que todas las correlaciones se han hallado entre variables de tipo entrada. Así pues, no es posible continuar con el análisis de productividad utilizando el conjunto de datos recibido. Esta limitación probablemente se deba al tamaño reducido de la muestra, con tan solo cuatro DMUs, lo que reduce las posibilidades de encontrar correlaciones entre variables para comprobar la isotonicidad. Por otro lado, al disponer de pocas DMUs, los potenciales modelos que pudieran plantearse deberían tener, como mucho y omitiendo las recomendaciones citadas en los casos de estudio anteriores, tres variables. Como posible solución a este contratiempo, se plantea a la organización participante aumentar la muestra añadiendo nuevas DMUs, pero ha resultado imposible disponer de estos participantes adicionales.

5.2.4.5. Generación y análisis de resultados

No ha sido posible generar resultados de productividad en este caso (ver el apartado anterior).

5.2.4.6. Conclusiones parciales

En este caso de estudio, las variables planteadas permitían, en un principio, llevar a cabo un análisis de productividad para el puesto de trabajo de Ingeniero de Software, pero no ha sido viable aplicar DEA. No obstante, de la muestra recibida pueden extraerse algunas conclusiones globales:

- **Las variables de interrupciones parecen medir el mismo elemento** (o los participantes no las distinguen), dado que la correlación entre ambas es alta (0,994) y estadísticamente significativa ($p < 0,01$).
- **Los trabajadores con mayor grado de teletrabajo disponen de menos conocimiento (experiencia) del proyecto**, dada la correlación alta (-0,980) y estadísticamente significativa ($p < 0,05$). Además, de forma casi estadísticamente significativa ($p = 0,06 > 0,05$), se encuentra una correlación alta positiva (0,940) entre la experiencia tecnológica y el grado de conocimiento del proyecto, por lo que **a mayor grado de experiencia en la tecnología empleada mayor conocimiento (experiencia) del proyecto**.
- **Se encuentran mayores interrupciones por dudas y preguntas en los trabajadores que desempeñan tareas menos difíciles**, dado que se ha encontrado una correlación alta y negativa (-0,991) y estadísticamente significativa ($p < 0,01$).
- Finalmente, tal y como se ha comentado en el caso de estudio 2, **la cantidad de variables que correlacionan se ha visto drásticamente reducida, hasta tal punto que no se han encontrado correlaciones entre ninguna variable de entrada y una de salida**. De este modo, y por el criterio de isotonicidad, **no ha sido posible aplicar DEA**.

De forma general, se puede concluir que antes de plantear futuros casos de estudio, aplicando DEA, se debería disponer del compromiso de las organizaciones

participantes a facilitar una cantidad mínima de participantes, ya que sin este compromiso la viabilidad de la metodología se ve comprometida y no es posible obtener resultados finales.

5.3. Conclusiones globales a los casos de estudio

Tras realizar cuatro casos de estudio es posible obtener una serie de conclusiones globales relativas a la utilización de DEA para medir la productividad en puestos de trabajadores vinculados con los proyectos de desarrollo y mantenimiento de software. Las conclusiones se presentan agrupadas por (1) limitaciones, (2) ventajas y desventajas con respecto a otras metodologías para medir la productividad, (3) recomendaciones para futuros casos de estudio, y (4) contraste de hipótesis de investigación.

5.3.1. Limitaciones

En primer lugar es necesario señalar que pese a haber dispuesto de cuatro organizaciones participantes, sólo en una de ellas ha sido posible llevar a cabo hasta el final la metodología propuesta. Esto limita en gran medida las conclusiones finales de esta investigación dado que no ha sido posible comparar la ejecución de la solución propuesta, llevada hasta el final, en dos organizaciones distintas. No obstante, dado que parte de la solución si ha podido ser llevada a cabo en las distintas organizaciones es posible obtener conclusiones relativas a dichas partes. De este modo, y en segundo lugar, la ejecución de la metodología basada en DEA se ha visto en gran medida limitada por la existencia de correlaciones entre variables de entrada y de salida (criterio de isotonicidad). Al no encontrarse al menos una correlación entre una entrada y algunas salidas o entre algunas entradas y una salida, no ha sido posible continuar con el método (ver 5.2.4 Caso de estudio 4) por lo que su eficacia es

limitada. Finalmente, el requisito de homogeneidad limita las posibilidades de realizar benchmarking entre resultados de organizaciones dado que este requisito sólo se cumple si los puestos de trabajo analizados son los mismos y los elementos diferenciadores están controlados.

5.3.2. Ventajas y desventajas

Las principales ventajas que pueden ser destacadas tras aplicar DEA como método de medición de productividad son las siguientes:

- Permite obtener un ranking interno de productividad. Los resultados de productividad son clasificados en comparación con los demás por lo que hay una clasificación de productividad para los trabajadores medidos.
- Permite conocer las personas de “referencia”. Además de la clasificación, DEA permite conocer personas referentes en cuanto a productividad para los trabajadores improductivos; es decir, su productividad en caso de mejorarla se aproximaría a la de dichas referencias.
- Permite conocer los elementos de mejora y cuanto se ha de mejorar cada uno. Este punto es importante, máxime en un entorno de ingeniería en el cuál la mejora es una piedra angular, dado que uno de los objetivos de medir es mejorar. En este sentido, DEA permite identificar qué variables deben reducirse o incrementarse, y en qué cantidad para que los trabajadores se conviertan en productivos. De este modo, los gestores tienen información para plantear propuestas de mejora basadas en información objetiva.
- Una vez aplicado, puede aplicarse de nuevo e ir incluyendo nuevas variables si se detectan nuevas correlaciones. Tras superar los requisitos de aplicación, el conjunto de variables puede utilizarse en posteriores mediciones para replicar el análisis con datos posteriores (o incluyéndolos para generar un resultado más global).

- Es posible aplicar varias medidas de productividad ya que a partir de un conjunto de variables, pueden plantearse varios modelos de productividad (ver 5.2.4 Caso de estudio 4) permitiendo combinar ambas medidas y resultados o tratarlas por separado, quedando a discreción de los objetivos perseguidos.

Las principales desventajas que pueden ser destacadas tras aplicar DEA como método de medición de productividad son las siguientes:

- No todas las variables planteadas inicialmente son válidas para aplicar DEA. Por el criterio de isotonicidad, la cantidad de variables utilizables se reduce, en muchos casos hasta paralizar el caso de estudio (ver 5.2.1 Caso 1 - Prueba piloto y 5.2.4 Caso de estudio 4).
- La cantidad de variables, y por lo tanto de datos a recoger, es alta y no garantiza que se pueda aplicar DEA. Esto representa una desventaja dado que las variables potencialmente medibles suelen ser variables que la organización no mide, lo que se traduce en un esfuerzo extra para llevar a cabo la medición de productividad. Si al aumento se le añade que un aumento de las variables y/o de los datos no garantiza la existencia de correlaciones que permitan cubrir el requisito de isotonicidad, se obtiene que el uso de DEA requiere de un alto esfuerzo para llevar a cabo la medición (por parte de los trabajadores y de los gestores). Esta desventaja es de especial influencia en momentos como los actuales en los que los gestores deben justificar muy bien los gastos de los proyectos. En el caso de la presente investigación, esta desventaja ha provocado que el Caso de estudio 4 se paralizará por no poder disponer de más datos con la esperanza de obtener correlaciones entre entradas y salidas.

5.3.3. Recomendaciones

En futuras aplicaciones de esta metodología se recomienda:

- Asegurar preliminarmente la disponibilidad de la organización para obtener de una cantidad de datos alta, así como el compromiso de llegar hasta el final del caso de estudio. Dadas las limitaciones y desventajas mencionadas anteriormente, se recomienda que la organización esté dispuesta a facilitar una cantidad de datos alta para que en caso de que, por algún impedimento inherente a los requisitos de aplicación de DEA, se paralice el caso de estudio, poder ampliar los datos y/o variables y continuar con el estudio.
- Disponer, en la medida de lo posible, de información que permita conocer correlaciones existentes para partir de una base fija de variables y aumentarla con nuevas variables. De este modo, en el peor de los casos los modelos a plantear sólo utilizaran las correlaciones ya conocidas, y en el resto de los casos será posible plantear nuevos modelos.
- Dado que es necesario comprobar correlaciones es posible combinar el uso de DEA con la realización de estudios que requieran dicha comprobación. Esta recomendación es casi exclusiva para el entorno científico dado que las organizaciones no suelen estar interesadas en analizar influencias ni efectos entre variables quedando este tipo de estudio relegado al ámbito científico. Además, dada la alta cantidad de información que se extrae, plantear la explotación de los datos en estudios paralelos cobra un mayor sentido, tanto por la reutilización de los datos como por el objetivo múltiple que puede conseguirse; máxime siendo la obtención de datos organizacionales una de las tareas más difíciles de la investigación en IS.

5.3.4. Contraste de hipótesis de investigación

Las dos últimas hipótesis planteaban:

***Hipótesis 5.** Las actuales medidas de productividad a nivel de puesto de trabajo en los proyectos de desarrollo de software tienen*

una eficacia limitada para medir la productividad real de los trabajadores.

Hipótesis 6. *Es posible medir de forma más eficaz la productividad de los puestos de trabajo en los proyectos de desarrollo de software con nuevas medidas que combinen varios elementos: entradas, salidas y factores.*

Mediante los casos de estudio realizados, es posible contrastar las hipótesis de forma cualitativa, ya que no ha sido posible obtener medidas de productividad de las organizaciones participantes que puedan ser vinculadas a los trabajadores participantes en cada caso de estudio, y por lo tanto comparadas con la metodología empleada. En primer lugar, con respecto a la H5 se puede asegurar que no se ha encontrado una medida que cubra toda la definición de productividad a nivel de puesto de trabajo en los puestos analizados por lo que es razonable pensar que la eficacia de las medidas actuales, incluida DEA, es limitada. Por otro lado, con respecto a H6, tiene sentido plantear, tras los casos de estudio realizados, que el uso de DEA es más eficaz para medir la productividad a nivel de puesto de trabajo que otras medidas de productividad basadas en líneas de código y/o puntos función exclusivamente, ya que permite incluir varias variables (tanto de entradas como de salidas) y puede ser ajustada a las necesidades de cada organización y puesto de trabajo.

Así pues, las hipótesis H5 y H6 han sido contrastadas con la información cualitativa disponible. Y se plantea, como línea futura de investigación, el contraste estadístico de las mismas.

6 Conclusiones y líneas futuras

6.1. Conclusiones

Tras la investigación realizada es posible llegar a ciertas conclusiones. En primer lugar, la presente tesis doctoral ejemplifica un proceso de generación de medidas de productividad para puestos de trabajo (en IS) que culmina con la aplicación de una medida, basada en una metodología ya existente (DEA), a un puesto de trabajo dentro de los proyectos de desarrollo y mantenimiento de software (ingeniero software).

Durante el proceso se ha realizado una revisión sistemática de la literatura cuyos resultados señalan que las medidas de productividad empleadas en IS a nivel de puesto de trabajo utilizan las mismas entradas y salidas que las empleadas en medidas de más alto nivel (proyecto, organización,...). No obstante, se ha encontrado una alternativa teórica basada en el uso de la cantidad de tareas completadas como unidad de medidas de salida. Esta alternativa no es práctica en la realidad dado que los trabajadores en IS (y en general los trabajadores del conocimiento) no realizan tareas idénticas y por lo tanto medir la productividad contando la cantidad de tareas realizadas no representaría una medida de productividad eficaz. De igual forma, las tareas no están estandarizadas ni existen unidades de obra medibles para cada una de los tipos de tarea realizables en los proyectos de desarrollo y mantenimiento de

software. Posteriormente, la realización de una investigación en dos fases, una cualitativa y otra cuantitativa, ha permitido profundizar en los objetivos de investigación y constatar que los puestos de trabajo relativos a los proyectos de desarrollo y mantenimiento de software emplean otros recursos, además del tiempo y el esfuerzo, y producen otros productos además de líneas de código y funcionalidad. Además, no todos los puestos utilizan los recursos y producen las salidas en igual cantidad y frecuencia. Por estos resultados se puede afirmar que el uso de medidas de productividad tales como SLOC/t o PF/t tiene una eficacia limitada (a la hora de medir la productividad en niveles inferiores a proyecto) y deben ser consideradas como medidas de productividad específicas, nunca medidas de productividad a nivel de puesto de trabajo (a no ser que el puesto en cuestión sólo requiera tiempo y produzca líneas de código o PF). Esta limitación es muy clara cuando se han contrastado estas medidas con las definiciones de productividad obtenidas en la fase cualitativa.

En segundo lugar, se concluye que para que una medida de productividad aplicada a puestos de trabajo, vinculados con los proyectos de desarrollo y mantenimiento de software, sea eficaz, deberá poder incluir varias entradas y varias salidas así como ser adaptable a las características de cada puesto de trabajo objeto de medición. Por esto, se ha decidido emplear la metodología DEA en varios casos de estudio, la cual permite medir la productividad en base a una serie de entradas y salidas, y conocer los puntos de mejora de los trabajadores menos eficientes.

En tercer lugar, y tras aplicar la metodología DEA en tres organizaciones (una de las organizaciones participantes no ha llegado a aportar datos) se concluye que esta metodología permite medir la productividad en los puestos de trabajo vinculados a los proyectos de desarrollo y mantenimiento de software. Además, se concluye que posee grandes ventajas (permite incluir varias entradas y varias salidas en una misma medida, aplicar varias medidas de productividad sobre un mismo conjunto de datos, señala los puntos de mejora, y es escalable y replicable) así como varias desventajas (limita y reduce la cantidad de datos recibidos, existe una alta probabilidad de no

alcanzar una medición de productividad con los datos obtenidos, y requiere un alto esfuerzo de recogida de información en caso de no disponer de sistemas digitales de gestión de trabajo).

En cuarto lugar, y para aplicar DEA, se debe destacar la dificultad que se ha encontrado de obtener datos reales de las organizaciones para los casos estudio. La dificultad fue en aumento desde los contactos iniciales hasta la confirmación de participación, así como la recepción de datos. Esta dificultad es un tema recurrente en las investigaciones en IS, por lo que para aumentar el beneficio de los datos obtenidos se recomienda, en la medida de lo posible, plantear varias investigaciones a partir de los conjuntos de datos a obtener. En la presente tesis, podrían haberse utilizado datos para llevar a cabo estudios que vincularan determinadas entradas con salidas (p. ej. motivación y satisfacción laboral, experiencia y eficiencia de empleo del tiempo) con un pequeño aumento de la cantidad de datos a recoger.

Finalmente, se concluye, tal y como afirmó Drucker, que la medición de la productividad de los trabajadores del conocimiento, en este caso, de los puestos de trabajo vinculados con los proyectos de desarrollo y mantenimiento software, es un reto, que requiere de futuras investigaciones. De forma general y para afrontar el reto, la presente tesis doctoral aporta un marco de trabajo para analizar y plantear medidas de productividad en distintos puestos de trabajo a partir de la fuente de conocimiento científico (mediante una revisión sistemática de la literatura) y a partir de los trabajadores que desempeñan los puestos de trabajo en cuestión (mediante una fase de investigación cualitativa y otra cuantitativa).

6.2. Líneas futuras

Dada la variedad de estudios realizados en la presente tesis doctoral es posible plantear líneas de investigación que amplían determinados estudios y profundicen en

áreas no exploradas. Así pues, a continuación se presentan un conjunto de líneas futuras agrupadas por el estudio preliminar que las motiva:

SLR

- Una mejora de la SLR realizada sería la inclusión de nuevas fuentes de información. Mediante esta ampliación, los resultados serían más globales. También podría modificarse la fecha de límite de búsqueda de modo que se incluyeran los estudios posteriores a la fecha de realización de la SLR (16/05/2011).

Cualitativo

- Para constatar los resultados obtenidos en esta fase, se plantea aplicar técnicas de grupo, p. ej. Delphi (Helmer, 1966), de modo que la opinión de todos los participantes estaría representada en la respuesta final, por lo que los resultados serán más objetivos y se reduciría la posibilidad de error.
- Dada la variedad de entradas y salidas detectadas, se plantea la posibilidad de construir una taxonomía de entradas y salidas del proceso productivo de desarrollo y mantenimiento de software que incluya potenciales medidas de cada uno de ellos.
- De igual forma, la construcción de una taxonomía que interrelacione factores incidentes en la productividad, que refleje el signo y grado de incidencia, y resuma todas las investigaciones llevadas hasta el momento representa otra línea de investigación.

Cuantitativo

- Con el objetivo de hacer los resultados más generalizados, se plantea el aumento de la muestra, con un posterior análisis de los datos. A su vez podría modificarse la forma de acceder a la muestra para aumentar la aleatoriedad de la muestra empleada y evitar los principales sesgos detectados.
- Pueden plantearse nuevas hipótesis de investigación, quizá más específicas a las que se han empleado en esta investigación para analizar los conceptos bajo

estudio. Por ejemplo, podría plantearse si el puesto de analista necesita más conocimiento funcional que el puesto de Jefe de Proyecto.

DEA

- Debido a que de los cuatro casos de estudios, sólo uno ha podido ser llevado hasta el final, se plantea realizar más estudios, con la misma metodología, en otras organizaciones. En esta línea, se recomienda disponer de un apoyo formal al caso de estudio (p. ej. un proyecto financiado de investigación).
- Una de las conclusiones de aplicar DEA (en cuatro organizaciones) es que la mayor dificultad es conseguir los datos. Esto es un problema recurrente en las demás investigaciones llevadas a cabo en el área de conocimiento de IS. Por ello, y para obtener un mayor beneficio de los datos obtenidos, se plantea como línea futura aplicar estudios de investigación paralelos a partir de los datos obtenidos para aplicar DEA. De este modo, el objetivo de investigación sería múltiple, y, quizá, los datos fueran de utilidad para los estudios no vinculados con DEA. Algunos ejemplos de preguntas de investigaciones que tiene sentido plantear a partir de los datos obtenidos para DEA son:
 - ¿Cómo afectan los diversos tipos experiencia a la productividad de los ingenieros software? (ver 5.2.1 Caso 1 - Prueba piloto).
 - ¿Sufren los puestos con mayor conocimiento funcional mayores interrupciones que los puestos más técnicos? (ver 5.2.2 Caso de estudio 2).
 - ¿El aumento de la motivación correlaciona con un aumento en la satisfacción laboral? (ver 5.2.2 Caso de estudio 2).
 - ¿Disminuye el conocimiento del proyecto a la vez que aumenta el teletrabajo? (ver 5.2.4 Caso de estudio 4).
 - ¿En qué medida un mayor conocimiento tecnológico permite adquirir el conocimiento del proyecto con mayor rapidez? (ver 5.2.4 Caso de estudio 4).

- En los casos de estudio no ha sido posible realizar una comparativa con otras medidas de productividad. Por ello, y para realizar un benchmarking de varias medidas de productividad, se plantea aplicar DEA en conjunto con otras medidas de productividad aplicadas a nivel de puesto de trabajo. El objetivo de esta línea de investigación es comparar la eficacia de las medidas analizadas así como comparar la dificultad de aplicación y el esfuerzo requerido para utilizarlas.

Global

- El marco de trabajo empleado en la presente investigación puede ser utilizado para estudiar otros puestos de trabajo, dentro y fuera de la IS. Esta línea, en momentos como los actuales en la que desde diversas partes se insta a la mejora de la productividad, así como a la vinculación de salarios a la productividad, cobra especial importancia. Sin una medida eficaz de productividad a nivel de trabajador es imposible evaluar la productividad del trabajador, y por lo tanto no es posible mejorarla ni vincularla con otros conceptos de la gestión de recursos humanos.

Bibliografía

- Abdoli, A., Shahrabi, J., & Heidary, J. (2011). Representing a Composing Fuzzy-DEA Model to Measure Knowledge Workers Productivity based upon their Efficiency and Cost Effectiveness. *Journal of Universal Computer Science*, 17(10), 1390-1411.
- Abufardeh, S., & Magel, K. (2010, 11-13 May). *The impact of global software cultural and linguistic aspects on Global Software Development process (GSD): Issues and challenges*. Paper presented at the 4th International Conference on New Trends in Information Science and Service Science (NISS).
- Acuña, S. T., Gómez, M., & Juristo, N. (2009). How do personality, team processes and task characteristics relate to job satisfaction and software quality? *Information and Software Technology*, 51(3), 627-639. doi: 10.1016/j.infsof.2008.08.006
- Acuña, S. T., Juristo, N., & Moreno, A. M. (2006). Emphasizing human capabilities in software development *IEEE Software*, 23(2), 94-101. doi: 10.1109/MS.2006.47
- Adam, K., Johanson, M., & Gravesen, I. (1995). *Service productivity: a vision or a search for a new outlook*. Paper presented at the 9th World Productivity Congress, Istanbul.
- Afzal, W., Torkar, R., & Feldt, R. (2009). A systematic review of search-based testing for non-functional system properties. *Information and Software Technology*, 51(6), 957-976. doi: 10.1016/j.infsof.2008.12.005

- Agarwal, R., & Ferratt, T. W. (1998, March 26-28). *Recruiting, retaining, and developing IT professionals: an empirically derived taxonomy of human resource practices*. Paper presented at the Computer Personnel Research Conference (SIGCPR'98), Boston, MA.
- Agarwal, R., & Ferratt, T. W. (2001). Crafting an HR strategy to meet the need for IT workers. *Communications of the ACM*, 44(7), 58-64. doi: 10.1145/379300.379314
- Agarwal, R., & Ferratt, T. W. (2002). Enduring practices for managing IT professionals. *Communications of the ACM*, 45(9), 73-79.
- Agrawal, M., & Chari, K. (2007). Software effort, quality, and cycle time: A study of CMM level 5 projects. *IEEE Transactions on Engineering Management*, 33(3), 145-156.
- Al-Darrab, I. A. (2000). Relationships between productivity, efficiency, utilization, and quality. *Work Study*, 49(3), 97-104. doi: 10.1108/00438020010318073
- Albrecht, A. J. (1979, October). *Measuring application development productivity*. Paper presented at the Joint SHARE/GUIDE/IBM Application Development Symposium, Monterey, CA.
- Albrecht, A. J., & Gaffney, J. E. (1983). Software Function, Source Lines of Code, and Development Effort Prediction: A Software Science Validation. *IEEE Transaction Software Engineering*, 9(6), 639-648. doi: 10.1109/tse.1983.235271
- Anda, B. C. D., Sjoberg, D. I. K., & Mockus, A. (2009). Variability and Reproducibility in Software Engineering: A Study of Four Companies that Developed the Same System. *IEEE Transaction Software Engineering*, 35(3), 407-429. doi: 10.1109/tse.2008.89
- Anselmo, D., & Ledgard, H. (2003). Measuring productivity in the software industry. *Communications of the ACM*, 46(11), 121-125.
- Asmild, M., Paradi, J. C., & Kulkarni, A. (2006). Using Data Envelopment Analysis in software development productivity measurement. *Software Process: Improvement and Practice*, 11(6), 561-572. doi: 10.1002/spip.298
- Baddoo, N., Hall, T., & Jagielska, D. (2006). Software developer motivation in a high maturity company: a case study. *Software Process: Improvement and Practice*, 11(3), 219-228.

- Banker, R. D., Charnes, A., & Cooper, W. W. (1984). Some Models for Estimating Technical and Scale Inefficiencies in Data Envelopment Analysis. *Management Science*, 30(9), 1078-1092.
- Banker, R. D., Datar, S. M., & Kemerer, C. (1987, December 6-9). *Factors Affecting Software Maintenance Productivity: An Exploratory Study*. Paper presented at the Eighth International Conference on Information Systems, Pittsburgh, PA.
- Banker, R. D., Datar, S. M., & Kemerer, C. F. (1991). A model to evaluate variable impacting the productivity of software maintenance projects. *Management Science*, 37(1), 1-18.
- Banker, R. D., & Kauffman, R. J. (1991). Reuse and Productivity in Integrated Computer-Aided Software Engineering: An Empirical Study. *MIS Quarterly*, 15(3), 375-401.
- Banker, R. D., & Kemerer, C. F. (1989). Scale Economies in New Software Development. *IEEE Transaction Software Engineering*, 15(10), 1199-1205. doi: 10.1109/tse.1989.559768
- Beecham, S., Baddoo, N., Hall, T., Robinson, H., & Sharp, H. (2008). Motivation in Software Engineering: A systematic literature review. *Information and Software Technology*, 50(9-10), 860-878. doi: 10.1016/j.infsof.2007.09.004
- Bellini, C., Pereira, R., & Becker, J. (2008). Measurement in software engineering: from the roadmap to the crossroads. *International Journal of Software Engineering and Knowledge Engineering*, 18(1), 37-64.
- Benediktsson, O., & Dalcher, D. (2004). Effort estimation in incremental software development. *IEE Proceedings-Software*, 150(6), 351-357. doi: 10.1049/ip-sen:20031056
- Berntsson-Svensson, R., & Aurum, A. (2006, September 21-22). *Successful software project and products: An empirical investigation*. Paper presented at the International Symposium on Empirical Software Engineering (ISESE'06), Rio de Janeiro, Brazil.
- Berry, L. M. (1997). *Psychology at work*. San Francisco, CA: McGraw-Hill Companies, Inc.
- Blackburn, J. D., Scudder, G. D., & Wassenhove, L. N. V. (1996). Improving Speed and Productivity of Software Development: A Global Survey of Software Developers. *IEEE Transactions on Software Engineering*, 22(12), 875-885. doi: 10.1109/32.553636

- Boehm, B. W. (1981a). An Experiment in Small-Scale Application Software Engineering. *IEEE Transactions on Software Engineering* 7(5), 482-493. doi: 10.1109/tse.1981.231110
- Boehm, B. W. (1981b). *Software Engineering Economics*. Upper Saddle River, NJ: Prentice Hall PTR.
- Boehm, B. W. (1987). Improving Software Productivity. *Computer*, 20(9), 43-57. doi: 10.1109/MC.1987.1663694
- Boehm, B. W. (2006). *A view of 20th and 21st century software engineering*. Paper presented at the 28th International Conference on Software Engineering, Shanghai, China.
- Boehm, B. W., Abts, C., Brown, A. W., Chulani, S., Clark, B. K., E. Horowitz, . . . Steece, B. (2000). *Software Cost Estimation with COCOMO II*. New York: Prentice-Hall.
- Boehm, B. W., Abts, C., & Chulani, S. (2000). Software development cost estimation approaches - A survey. *Annals of Software Engineering*, 10(1), 177-205.
- Boehm, B. W., & Ross, R. (1989). Theory-W Software Project Management Principles and Examples. *IEEE Transactions on Software Engineering*, 15(7), 902-916. doi: 10.1109/32.29489
- Bok, H. S., & Raman, K. S. (2000). Software engineering productivity measurement using function points: a case study. *Journal of Information Technology*, 15(1), 79-90. doi: 10.1080/026839600344429
- Boussofiane, A., Dyson, R. G., & Thanassoulis, E. (1991). Applied data envelopment analysis. *European Journal of Operations Research*, 52(1), 1-15.
- Briand, L. C., Morasca, S., & Basili, V. R. (2002). An Operational Process for Goal-Driven Definition of Measures. *IEEE Transactions on Software Engineering*, 28(12), 1106-1125. doi: 10.1109/tse.2002.1158285
- Brooks Jr., F. P. (1985). *The Mythical Man-Month: Essays on Software Engineering*. Reading, MA: Addison-Wesley Professional.
- Burn, J. M., Couger, J. D., Ma, L., & Tompkins, H. (1991, January 08-11). *Motivating IT professionals. The Hong Kong challenge*. Paper presented at the Twenty-Fourth Annual Hawaii International Conference on System Sciences.
- Carmel, E. (1999). *Global software teams: collaborating across borders and time zones*. Upper Saddle River, NJ: Prentice Hall.

- Carsten, J. M., & Spector, P. E. (1987). Unemployment, job satisfaction, and employee turnover: A meta-analytic test of the Muchinsky model. *Journal of Applied Psychology*, 72(3), 374-381.
- Ciolkowski, M., Laitenberger, O., Vegas, S., & Biffl, S. (2003). Practical Experiences in the Design and Conduct of Surveys in Empirical Software Engineering. *Lecture Notes in Computer Science*, 2765, 104-128. doi: 10.1007/978-3-540-45143-3_7
- Cockburn, A., & Highsmith, J. (2001). Agile software development, the people factor. *Computer*, 34(11), 131-133. doi: 10.1109/2.963450
- Cole, A. (1995). Runaway projects – cause and effects. *Software World*, 26(3), 3-5.
- Colombo, G., Mosca, A., & Sartori, F. (2007). Towards the design of intelligent CAD systems: an ontological approach. *Advanced Engineering Informatics*, 21(2), 153-168.
- Colomo-Palacios, R., Cabezas-Isla, F., García-Crespo, Á., & Soto-Acosta, P. (2010). Generic Competences for the IT Knowledge Workers: A Study from the Field. In M. D. Lytras, P. Ordonez De Pablos, A. Ziderman, A. Roulstone, H. Maurer & J. B. Imber (Eds.), *Knowledge Management, Information Systems, E-Learning, and Sustainability Research* (Vol. 111, pp. 1-7): Springer Berlin Heidelberg.
- Colomo-Palacios, R., Tovar-Caro, E., García-Crespo, Á., & Gómez-Berbís, J. M. (2010). Identifying Technical Competences of IT Professionals: The Case of Software Engineers. *International Journal of Human Capital and Information Technology Professionals*, 1(1), 31-43.
- Cook, W. D., & Seiford, L. M. (2009). Data envelopment analysis (DEA) – Thirty years on. *European Journal of Operational Research*, 192(1), 1-17. doi: 10.1016/j.ejor.2008.01.032
- Cooper, W. W., Seiford, L. M., & Tone, K. (2006). *Introduction to Data Envelopment Analysis and its Uses: With DEA-Solver Software and References*. New York: Springer.
- Couger, J. D., Adelsberger, H., Borovits, I., Zviran, M., & Motiwalla, J. (1990). Commonalities in motivating environments for programmer/analysts in Austria, Israel, Singapore, and the U.S.A. *Information & Management*, 18(1), 41-46.

- Craig, C. E., & Harris, R. C. (1973). Total productivity measurement at the firm level. *Sloan Management Review*, 14(3), 13-29.
- Crepeau, R. G., Crook, C. W., Goslar, M. D., & McMurtrey, M. E. (1992). Career Anchors of Information Systems Personnel. *Journal of Management Information Systems*, 9(2), 145-160.
- Curtis, B., Krasner, H., & Iscoe, N. (1988). A field study of the software design process for large systems. *Communications of the ACM*, 31(11), 1268-1287.
- Cusumano, M. A., & Kemerer, C. F. (1990). A quantitative analysis of U.S. and Japanese practice and performance in software development. *Management Science*, 36(11), 1384-1406. doi: 10.1287/mnsc.36.11.1384
- Charnes, A., Cooper, W. W., Lewin, A., & Seiford, L. M. (1994). *Data Envelopment Analysis: Theory, Methodology and Application*. Boston, MA: Kluwer Academic Publishers.
- Charnes, A., Cooper, W. W., & Rhodes, E. (1978). Measuring the efficiency of decision making units. *European Journal of Operations Research*, 2(6), 429-444.
- Chatzigeorgiou, A., & Stiakakis, E. (2013). Combining metrics for software evolution assessment by means of Data Envelopment Analysis. *Journal of Software: Evolution and Process*, 25(3), 304-324. doi: 10.1002/smr.584
- Chatzoglou, P. D., & Macaulay, L. A. (1997). The importance of human factors in planning the requirements capture stage of a project. *International Journal of Project Management*, 15(1), 39-53.
- Cheikhi, L., Al-Qutaish, R. E., & Idri, A. (2012). Software Productivity: Harmonization in ISO/IEEE Software Engineering Standards. *Journal of Software*, 7(2), 462-470. doi: 10.4304/jsw.7.2.462-470
- Chen, T. (1997). A measurement of the resource utilization efficiency of university libraries. *International Journal of Production Economics*, 53(1), 71-80.
- Chrysler, E. (1978). Some basic determinants of computer programming productivity. *Communications of the ACM*, 21(6), 472-483. doi: 10.1145/359511.359523
- Chung, S.-H., Lee, A. H.-I., Kang, H.-Y., & Lai, C.-W. (2008). A DEA window analysis on the product family mix selection for a semiconductor fabricator. *Expert Systems with Applications*, 35(1-2), 379-388. doi: 10.1016/j.eswa.2007.07.011

- Dalcher, D. (2006). Supporting software development: enhancing productivity, management and control. *Software Process: Improvement and Practice*, 11(6), 557-559. doi: 10.1002/spip.317
- Dale, C. J., & van der Zee, H. (1992). Software productivity metrics: who needs them? *Information and Software Technology*, 34(11), 731-738. doi: [http://dx.doi.org/10.1016/0950-5849\(92\)90168-O](http://dx.doi.org/10.1016/0950-5849(92)90168-O)
- Davenport, T. H. (2005). *Thinking for a living: How to get better performance and results from knowledge workers*. Boston, MA: Harvard Business School Press.
- Davenport, T. H., & Prusak, L. (2000). *Working Knowledge: How Organizations Manage What They Know*. Boston, MA: Harvard Business School Press.
- de Aquino Junior, G. S., & De Lemos Meira, S. R. (2009). *Towards Effective Productivity Measurement in Software Projects*. Paper presented at the Fourth International Conference on Software Engineering Advances (ACSEA'09), Porto, Brazil.
- de Barros Sampaio, S. C., Barros, E. A., de Aquino Junior, G. S., e Silva, M. J. C., & de Lemos Meira, S. R. (2010). *A Review of Productivity Factors and Strategies on Software Development*. Paper presented at the Fifth International Conference on Software Engineering Advances (ICSEA '10), Nize, France.
- de Souza Carvalho, W. C., Rosa, P. F., dos Santos Soares, M., Teixeira da Cunha Junior, M. A., & Buiatte, L. C. (2011, 9-11 Nov. 2011). *A Comparative Analysis of the Agile and Traditional Software Development Processes Productivity*. Paper presented at the 30th International Conference of the Chilean Computer Science Society (SCCC).
- DeMarco, T. (1986). *Controlling Software Projects: Management, Measurement, and Estimates*. Upper Saddle River, NJ: Prentice Hall PTR.
- DeMarco, T., & Lister, T. (1985, August 28-30). *Programmer Performance and the Effects of the Workplace*. Paper presented at the 8th International Conference on Software Engineering, London, UK.
- DeMarco, T., & Lister, T. (1987). *Peopleware: productive projects and teams*. New York: Dorset House Publishing Co., Inc.
- DeMarco, T., & Lister, T. (1999). *Peopleware: productive projects and teams* (2nd ed.). New York: Dorset House Publishing Co., Inc.
- Denzin, N. K., & Lincoln, Y. S. (2011). *The SAGE Handbook of Qualitative Research*. Thousand Oaks, CA: Sage Publications.

- Desharnais, J. M., & April, A. (2010). *Software maintenance productivity and maturity*. Paper presented at the 11th International Conference on Product Focused Software, Limerick, Ireland.
- Desharnais, J. M., Yıldızoğlu, G., April, A., & Abran, A. (2013). Building Productivity Models for Small Enhancements. *Journal of Software Engineering and Applications*, 6(3), 121-130. doi: 10.4236/jsea.2013.63016
- Drucker, P. (1959). *The Landmarks of Tomorrow*. New York: Harper & Row.
- Drucker, P. (1999). Knowledge-Worker Productivity: The Biggest Challenge. *California management review*, 41(2), 79-85.
- Dyba, T., & Dingsoyr, T. (2009). What Do We Know about Agile Software Development? *IEEE Software*, 26(5), 6-9. doi: 10.1109/MS.2009.145
- Eisenberger, R., Rhoades, L., & Cameron, J. (1999). Does pay for performance increase or decrease perceived self-determination and intrinsic motivation? *Journal of personality and social psychology* 77(5), 1026-1040.
- Erceg-Hurn, D. M., & Mirosevich, V. M. (2008). Modern robust statistical methods: An easy way to maximize the accuracy and power of your research. *American Psychologist*, 63(7), 591-601. doi: 10.1037/0003-066X.63.7.591
- Erne, R. (2011). What is Productivity in Knowledge Work? - A Cross-industrial View. *Journal of Universal Computer Science*, 17(10), 1367-1389.
- Farrell, M. J. (1957). The measurement of productive efficiency. *Journal of the Royal Statistical Society*, 120(3), 253-281.
- Farris, J. A., Groesbeck, R. L., Van Aken, E. M., & Letens, G. (2006). Evaluating the relative performance of engineering design projects: A case study using data envelopment analysis. *IEEE Transactions on Engineering Management*, 53(3), 471-482.
- Fenton, N. E., & Neil, M. (1999). Software metrics: success, failures and new directions. *Journal of Systems and Software*, 47(2-3), 149-157. doi: 10.1016/S0164-1212(99)00035-7
- Fernández-Ríos, M. (1995). *Análisis y descripción de puestos de trabajo*. Madrid: Diaz de Santos.
- Ferratt, T. W., & Short, L. E. (1986). Are information systems people different: an investigation of motivational differences. *MIS Quarterly*, 10(4), 377-387.

- Fink, A. (2005). *Conducting Research Literature Reviews. From the Internet to Paper*. Thousand Oaks, CA: Sage Publication, Inc.
- Finnie, G. R., Wittig, G. E., & Petkov, D. I. (1993). Prioritizing software development productivity factors using the analytic hierarchy process. *Journal of Systems and Software*, 22(2), 129-139. doi: 10.1016/0164-1212(93)90091-b
- Fitzgerald, L., & Moon, P. (1996). *Performance measurement in service industries: making it work*. London: Cima.
- Foddy, W. H. (1994). *Constructing questions for interviews and questionnaires: Theory and practice in social research*. Cambridge University Press.
- França, A. C. C., & da Silva, F. Q. B. (2010). *Designing motivation strategies for software engineering teams: an empirical study*. Paper presented at the Workshop on Cooperative and Human Aspects of Software Engineering (ICSE'10), Cape Town, South Africa.
- Frangos, S. A. (1997). Motivated humans for reliable software products. *Microprocessors and Microsystems*, 21(10), 605-610.
- Gaffney, J. (1989). Software reuse--key to enhanced productivity: some quantitative models. *Information and Software Technology*, 31(5), 258-267.
- Gambill, S. E., Clark, W. J., & Wilkes, R. B. (2000). Toward a holistic model of task design for IS professionals. *Information and Management*, 37(5), 217-228.
- Garcia-Crespo, A., Colomo-Palacios, R., Gomez-Berbis, J. M., & Tovar-Caro, E. (2008). The IT Crowd: Are We Stereotypes? *IT Professional*, 10(6), 24-27.
- Garden, A. (1988). Maintaining the spirit of excitement in growing companies. *SIGCPR Computer Personnel*, 11(4), 10-12. doi: 10.1145/54127.54130
- Garza, A. I., Lunce, S. E., & Maniam, B. (2003). *Career anchors of Hispanic information systems professionals*. Paper presented at the Annual Meeting of the Decision Sciences Institute.
- Ghapanchi, A. H., & Aurum, A. (2012). The impact of project capabilities on project performance: Case of open source software projects. *International Journal of Project Management*, 30(4), 407-417. doi: 10.1016/j.ijproman.2011.10.002
- Ghobadian, A., & Husband, T. (1990). Measuring total productivity using production functions. *International Journal of Production Research* 28(8), 1435-1446. doi: 10.1080/00207549008942803

- Glass, R. L. (1998). Reuse: What's Wrong with This Picture? *IEEE Software*, 15(2), 57-59. doi: 10.1109/52.663785
- Golembiewski, R. T. (2000). *Handbook of Organizational Behavior* (2th ed.): CRC Press.
- Gómez, O., Oktaba, H., Piattini, M., & García, F. (2008). A Systematic Review Measurement in Software Engineering: State-of-the-Art in Measures. In J. Filipe, B. Shishkov & M. Helfert (Eds.), *Software and Data Technologies* (Vol. 10, pp. 165-176): Springer Berlin Heidelberg.
- Gorla, N., & Lam, Y. W. (2004). Who should work with whom?: building effective software project teams. *Communications of the ACM*, 47(6), 79-82. doi: 10.1145/990680.990684
- Grady, R. B., & Caswell, D. L. (1987). *Software Metrics: Establishing a Company-Wide Program*. New York: Prentice-Hall.
- Griffell-Tatjé, E., & Lovell, C. A. K. (1999). Profits and Productivity. *Management Science*, 45(9), 1177-1193. doi: 10.1287/mnsc.45.9.1177
- Grönroos, C., & Ojasalo, K. (2004). Service productivity: Towards a conceptualization of the transformation of inputs into economic results in services. *Journal of Business Research*, 57(4), 414-423.
- GUFPI-ISMA, S. (2012). Tassonomia dei Fattori d'Impatto della Produttività dei Progetti Software: GUFPI-ISMA.
- Gummesson, E. (1992). Quality dimensions: what to measure in service organizations. In T. A. Swartz, D. E. Bowen & S. W. Brown (Eds.), *Advances in services marketing and management* (pp. 64-78). Greenwich, CT: JAI Press.
- Gummesson, E. (1995). Service productivity: a blasphemous approach. In E. Gummesson (Ed.), *Quality, productivity and profitability in service operations (conference papers from the QP&P Research Program 1992-1994)* (pp. 8-22). Stockholm: University of Stockholm, Department of Business Administration.
- Gupta, A. (1995). Productivity measurement in service operations: a case study from the health-care environment. *Managing Service Quality*, 5(5), 31-31. doi: 10.1108/09604529510100468
- Guzmán, J. G., Ramos, J. S., Seco, A. A., & Esteban, A. S. (2011). Success Factors for the Management of Global Virtual Teams for Software Development. *International Journal of Human Capital and Information Technology Professionals*, 2(2), 48-59. doi: 10.4018/jhcitp.2011040105

- Guzzo, R. A. (1988). Productivity research: Reviewing psychological and economic perspectives. In J. P. Campbell & R. J. Campbell (Eds.), *Productivity in Organizations* (pp. 63-81). San Francisco, CA: Jossey-Bass.
- Hackman, J. R., & Oldham, G. R. (1980). *Work Redesign*. Reading, MA: Addison-Wesley.
- Hall, T., Beecham, S., Verner, J., & Wilson, D. (2008). *The impact of staff turnover on software projects: the importance of understanding what makes software practitioners tick*. Paper presented at the ACM SIGMIS CPR conference on Computer personnel doctoral consortium and research, Charlottesville, VA, USA.
- Hall, T., Sharp, H., Beecham, S., Baddoo, N., & Robinson, H. (2008). What Do We Know about Developer Motivation? . *IEEE Software*, 25(4), 92-94. doi: 10.1109/MS.2008.105
- Hayes, W. (1999). *Research Synthesis in Software Engineering: A Case for Meta-Analysis*. Paper presented at the 6th International Symposium on Software Metrics.
- Helmer, O. (1966). *Social Technology*. NY: Basic Books.
- Herbsleb, J. D., & Moitra, D. (2001). Global Software Development. *IEEE Software*, 18(2), 16-20. doi: 10.1109/52.914732
- Hernández-López, A., Colomo-Palacios, R., & García-Crespo, Á. (2012, 23/06/2012). *Productivity in software engineering: a study of its meanings for practitioners*. Paper presented at the Second European Workshop on Computing and ICT Professionalism (EWCIP 2012), Madrid, Spain.
- Hernández-López, A., Colomo-Palacios, R., & García-Crespo, Á. (2013). Software Engineering Job Productivity - A Systematic Review. *International Journal of Software Engineering and Knowledge Engineering*, 23(3), 387-406. doi: 10.1142/S0218194013500125
- Hernández-López, A., Colomo-Palacios, R., García-Crespo, Á., & Soto-Acosta, P. (2010). Team Software Process in GSD Teams: A study of new work practices and models. *International Journal of Human Capital and Information Technology Professionals*, 1(3), 32-53. doi: 10.4018/jhcitp.2010070103
- Herzberg, F., Mausner, B., & Snyderman, B. B. (1959). *The motivation to work*. New York: John Wiley.

- Hill, E. J., Miller, B. C., Weiner, S. P., & Colihan, J. (1998). Influences of the virtual office on aspects of work and work/life balance. *Personnel Psychology*, 51, 667-683.
- Horibe, F. (1999). *Managing Knowledge Workers*. New York: John Wiley.
- Houkesa, I., Janssen, P. P. M., de Jongeb, J., & Nijhuisa, F. J. N. (2001). Specific relationships between work characteristics and intrinsic work motivation, burnout and turnover intention: A multi-sample analysis. *European Journal of Work and Organizational Psychology*, 10(1), 1-23.
- Hove, S. E., & Anda, B. (2005). *Experiences from Conducting Semi-structured Interviews in Empirical Software Engineering Research*. Paper presented at the Proceedings of the 11th IEEE International Software Metrics Symposium.
- Hsu, J. S.-C., Chen, H.-G., Jiang, J., & Klein, G. (2010). The Role of User Review on Information System Project Outcomes: A Control Theory Perspective *International Journal of Information Technology Project Management*, 1(1), 1-14. doi: 10.4018/jitpm.2010100201
- Hsu, M. K., Chen, H. G., Jiang, J. J., & Klein, G. (2003). Career satisfaction for managerial and technical anchored IS personnel in later career stages. *SIGMIS Database*, 34(4), 64-72. doi: 10.1145/957758.957766
- Hulin, C. L. (1991). Adaptation, persistence, and commitment in organizations. In M. D. Dunnette & L. M. Hough (Eds.), *Handbook of industrial and organizational psychology Vol. 2* (pp. 445–505). Palo Alto, CA: Consulting Psychologists Press.
- Humphrey, W. S. (1997). *Introduction to the Personal Software Process*. New York: Addison-Wesley.
- Igbaria, M., Meredith, G., & Smith, D. C. (1994). Predictors of intention of IS professionals to stay with the organization in South Africa. *Information & Management*, 26(5), 245-256. doi: 10.1016/0378-7206(94)90063-9
- Igbaria, M., Meredith, G., & Smith, D. C. (1995). Career orientations of information systems employees in South Africa. *Journal of Strategic Information Systems*, 4(4), 319-340.
- INE. (2011). Indicadores del Sector TIC - Año 2009 *Indicadores del Sector TIC*: Instituto Nacional de Estadística.
- Informática, C. S. d. (2000). Métrica Versión 3: Ministerio de Administraciones Públicas.

ISO. (2006). ISO/IEC 14764.

Javed, T., Manzil-e-Maqsood, & Durrani, Q. S. (2004). A survey to examine the effect of team communication on job satisfaction in software industry. *ACM SIGSOFT Software Engineering Notes*, 29(2), 6-6. doi: 10.1145/979743.979760

Jefferys, J., Hausberger, S., & Lindblad, G. (1954). *Productivity in the distributive trade in Europe: wholesale and retail aspects*: Organisation for European Economic Co-operation.

Johns, G. (1997). Contemporary research on absence from work: Correlates, causes, and consequences. In C. L. C. I. T. Robertson (Ed.), *International review of industrial and organizational psychology* (Vol. 12, pp. 115-173). Chichester, UK: Wiley.

Jones, C. (1981). *Programmer Productivity: Issues for the Eighties*. Los Alamitos, CA: IEEE Computer Soc. Press.

Jones, C. (1995). How Office Space Affects Programming Productivity. *Computer*, 28(1), 76. doi: 10.1109/2.362626

Jones, C. (2000). *Software Assessments, Benchmarks, and Best Practices*. Reading, MA: Addison-Wesley.

Jordan, E., & Whiteley, A. M. (1994). *HRM practices in information technology management*. Paper presented at the Computer Personnel Research Conference (SIGCPR'94), Alexandria, VA.

Jørgensen, M., & Shepperd, M. (2007). A Systematic Review of Software Development Cost Estimation Studies. *IEEE Transactions on Software Engineering*, 33(1), 33-53. doi: 10.1109/tse.2007.3

Judge, T. A., & Church, A. H. (2000). Job satisfaction: Research and practice. In C. L. Cooper & E. A. Locke (Eds.), *Industrial and organizational psychology: Linking theory with practice* (pp. 166-198). Oxford: Blackwell.

Judge, T. A., Heller, D., & Mount, M. K. (2002). Five-factor model of personality and job satisfaction: A meta-analysis. *Journal of Applied Psychology* 87(3), 530-541.

Judge, T. A., Piccolo, R. F., Podsakoff, N. P., & Shaw, J. C. (2010). The relationship between pay and job satisfaction: A meta-analysis of the literature. *Bruce L. Rich. Journal of Vocational Behavior*, 77(2), 157-167.

- Judge, T. A., Thoresen, C. J., Bono, J. E., & Patton, G. K. (2001). The job satisfaction–job performance relationship: A qualitative and quantitative review. *Psychological Bulletin*, 127(3), 376-407. doi: 10.1037/0033-2909.127.3.376
- Keselman, H. J., Huberty, C. J., Lix, L. M., Olejnik, S., Cribbie, R. A., Donahue, B., . . . Levin, J. R. (1998). Statistical practices of educational researchers: An analysis of their ANOVA, MANOVA, and ANCOVA analyses. *Review of Educational Research*, 68(3), 350-386.
- Kieburtz, R. B., McKinney, L., Bell, J. M., Hook, J., Kotov, A., Lewis, J., . . . Walton, L. (1996, March 25-29). *A software engineering experiment in software component generation*. Paper presented at the 18th International Conference on Software Engineering, Berlin , Germany
- Kitchenham, B. A. (2004). Procedures for Performing Systematic Reviews. Keele , UK: Software Engineering Group, Keele University.
- Kitchenham, B. A. (2007). Guidelines for performing Systematic Literature Reviews in Software Engineering (Vol. Version 2.3.): Keele University and University of Durham.
- Kitchenham, B. A. (2010). What's up with software metrics? - A preliminary mapping study. *Journal of Systems and Software*, 83(1), 37-51. doi: 10.1016/j.jss.2009.06.041
- Kitchenham, B. A., Hughes, R. T., & Linkman, S. G. (2001). Modeling Software Measurement Data. *IEEE Transactions on Software Engineering*, 27(9), 788-804. doi: 10.1109/32.950316
- Kitchenham, B. A., & Mendes, E. (2004). Software Productivity Measurement Using Multiple Size Measures. *IEEE Transactions on Software Engineering*, 30(12), 1023-1035. doi: 10.1109/TSE.2004.104
- Kitchenham, B. A., Pfleeger, S. L., Pickard, L. M., Jones, P. W., Hoaglin, D. C., El Emam, K., & Rosenberg, J. (2002). Preliminary guidelines for empirical research in software engineering. *IEEE Transactions on Software Engineering*, 28(8), 721-734. doi: 10.1109/tse.2002.1027796
- Klenke, K., & Kievit, K.-A. (1992). *Predictors of leadership style, organizational commitment and turnover of information systems professionals*. Paper presented at the Computer Personnel Research Conference (SIGCPR'92), Cincinnati, Ohio.

- Kling, R., & Scacchi, W. (1982). The web of computing: Computer technology as social organization. *Advances in Computers*, 21, 1-90.
- Koch, M. J., & McGrath, R. G. (1996). Improving labor productivity: Human resource management policies do matter. *Strategic Management Journal*, 17(5), 335-354.
- Koskinen, K. U. (2008). Boundary brokering as a promoting factor in competence sharing in a project work context. *International Journal of Project Organisation and Management*, 1(1), 119-132. doi: 10.1504/IJPOM.2008.020032
- Koss, E., & Lewis, D. A. (1993). Productivity or efficiency— Measuring what we really want. *National Productivity Review*, 12(2), 273-284. doi: 10.1002/npr.4040120212
- Krishnan, M. S., Kriebel, C. H., Kekre, S., & Mukhopadhyay, T. (2000). An Empirical Analysis of Productivity and Quality in Software Products. *Management Science*, 46(6), 745-759.
- Kruskal, W. H., & Wallis, W. A. (1952). Use of ranks in one-criterion variance analysis. *Journal of the American Statistical Association*, 47, 583-621.
- Kvale, S. (2008). *InterViews: Learning the Craft of Qualitative Research Interviewing*. Thousand Oaks, CA: Sage Publication.
- Lakhanpal, B. (1993). Understanding the factors influencing the performance of software development groups: An exploratory group-level analysis. *Information and Software Technology*, 35(8), 468-473.
- Laranjeira, L. A. (1990). Software Size Estimation of Object-Oriented Systems. *IEEE Transactions on Software Engineering*, 16(5), 510-522. doi: 10.1109/32.52774
- LeDuc, A. L. J. (1980). Motivation of programmers. *SIGMIS Database*, 11(4), 4-12.
- Lee, M. S. D., Trauth, E. M., & Farwell, D. (1995). Critical Skills and Knowledge Requirements of IS Professionals: A Joint Academic/Industry Investigation. *MIS Quarterly*, 19(3), 313-340.
- Levene, H. I., Harold Hotelling, et alia. . pp. . (1960). Robust tests for equality of variances. In I. Olkin, S. G. Ghurye, W. Hoeffding, W. G. Madow & H. B. Mann (Eds.), *Contributions to Probability and Statistics: Essays in Honor of Harold Hotelling* (pp. 278-292). Stanford, CA: Stanford University Press.

- Li, Y., Tan, C.-H., Teo, H.-H., & Mattar, A. T. (2006). *Motivating open source software developers: influence of transformational and transactional leaderships*. Paper presented at the Computer Personnel Research Conference (SIGCPR'06), Claremont, CA.
- Likert, R. (1932). A Technique for the Measurement of Attitudes. *Archives of Psychology*, 140(1-55).
- Linberg, K. R. (1999). Software developer perceptions about software project failure: a case study. *Journal of Systems and Software*, 49(2-3), 177-192.
- Linna, P., Karttunen, E., & Jaakkola, H. (2011, 23-27 May). *Software engineering companies' multicultural education*. Paper presented at the International Convention MIPRO.
- Liping, D., Qiusong, Y., Sun, L., Tong, J., & Wang, Y. (2005). *Evaluation of the capability of personal software process based on data envelopment analysis*. Paper presented at the International Conference on Unifying the Software Process Spectrum, Beijing, China.
- Litecky, A., Aken, C., Ahmad, A., & Nelson, J. (2010). Mining for Computing Jobs. *IEEE Software*, 27(1), 78-85. doi: 10.1109/ms.2009.150
- Liu, J. S., Lu, L. Y. Y., Lu, W.-M., & Lin, B. J. Y. (2013). Data envelopment analysis 1978–2010: A citation-based literature survey. *Omega*, 41(1), 3-15. doi: 10.1016/j.omega.2010.12.006
- Liu, W., Meng, W., Li, X., & Zhang, D. (2010). DEA models with undesirable inputs and outputs. *Annals of Operations Research*, 173(1), 177-194. doi: 10.1007/s10479-009-0587-3
- Locke, E. A. (1969). What is job satisfaction? *Organizational Behavior and Human Performance*, 4(4), 309-336. doi: 10.1016/0030-5073(69)90013-0
- Locke, E. A. (1976). The nature and causes of job satisfaction. In M. D. Dunnette (Ed.), *Handbook of industrial and organizational psychology* (pp. 1297-1349). Chicago, IL: Rand McNally.
- Locke, E. A., Feren, D. B., McCaleb, V. M., Shaw, K. N., & Denny, A. T. (1980). The relative effectiveness of four methods of motivating employee performance. In K. D. Duncan, M. M. Gruenberg & D. Wallis (Eds.), *Changes in working life* (pp. 363-388). New York: Wiley.

- Lodahl, T. M., & Kejnar, M. (1965). The definition and measurement of job involvement. *Journal of Applied Psychology*, 49(1), 24-33. doi: 10.1037/h0021692
- London, M. (1983). Toward a Theory of Career Motivation. *The Academy of Management Review*, 8(4), 620-630.
- López-Martín, C., Chavoya-Peña, A., & Meda-Campaña, M. E. (2012, 16-19 July, 2012). *Comparison of Software Development Productivity Based on Object-Oriented Programming Languages*. Paper presented at the International Conference on Software Engineering Research & Practice (SERP 2012), Las Vegas, NV.
- Low, G. C., & Jeffery, D. R. (1990). Function Points in the Estimation and Evaluation of the Software Process. *IEEE Transactions on Software Engineering*, 16(1), 64-71. doi: 10.1109/32.44364
- Lowry, M. R. (2010). *Towards predictive models of technology impact on software design productivity*. Paper presented at the FSE/SDP Workshop on Future of Software Engineering Research, Santa Fe, NM.
- MacCormack, A., Kemerer, C. F., Cusumano, M., & Crandall, B. (2003). Trade-offs between Productivity and Quality in Selecting Software Development Practices. *IEEE Software*, 20(5), 78-85. doi: 10.1109/ms.2003.1231158
- Mahmood, M. A., Pettingell, K. J., & Shaskevich, A. I. (1996). Measuring Productivity of Software Projects: A Data Envelopment Analysis Approach. *Decision Sciences*, 27(1), 57-80. doi: 10.1111/j.1540-5915.1996.tb00843.x
- Mak, B. L., & Sockel, H. (2001). A confirmatory factor analysis of IS employee motivation and retention. *Information & Management*, 38(5), 265-276.
- Martin, J. (1981). *Application Development Without Programmers*: Prentice Hall.
- Maslow, A. H. (1943). A theory of human motivation. . *Psychological Review*, 50(4), 370-396. doi: 10.1037/h0054346
- Maslow, A. H. (1954). *Motivation and Personality*: Harper & Row.
- Massey, F. J. (1951). The Kolmogorov-Smirnov Test for Goodness of Fit. *Journal of the American Statistical Association*, 46(253), 68-78. doi: 10.1080/01621459.1951.10500769

- Mata-Toledo, R. A., & Unger, E. A. (1985). Another look at motivating data processing professionals. *ACM SIGCPR Computer Personnel*, 10(1), 1-7. doi: 10.1145/1036354.1036355
- Maurer, F., & Martel, S. (2002). *On the Productivity of Agile Software Practices: An Industrial Case Study*. Paper presented at the International Workshop on Economics-Driven Software Engineering Research (EDSER).
- Maxwell, K. D., & Forselius, P. (2000). Benchmarking Software-Development Productivity. *IEEE Software*, 17(1), 80-88. doi: 10.1109/52.820015
- Maxwell, K. D., Wassenhove, L. V., & Dutta, S. (1996). Software Development Productivity of European Space, Military, and Industrial Applications. *IEEE Transactions on Software Engineering*, 22(10), 706-718. doi: 10.1109/32.544349
- McIver, J. P., & Carmines, E. G. (1981). *Unidimensional Scaling, Quantitative Applications in the Social Sciences*.: Sage Publications.
- McLean, E. R., Smits, S. J., & Tanner, J. R. (1996). The importance of salary on job and career attitudes of information systems professionals. *Information & Management*, 30(6), 291-299. doi: 10.1016/S0378-7206(96)01059-2
- Mei-Chi, L., Hao-Chen, H., & Wei-Kang, W. (2011). Designing a knowledge-based system for benchmarking: A DEA approach. *Knowledge-Based Systems*, 24(5), 662-671. doi: 10.1016/j.knosys.2011.02.006
- Melnik, G., & Maurer, F. (2006). *Comparative Analysis of Job Satisfaction in Agile and Non-agile Software Development Teams*. Paper presented at the 7th International Conference Extreme Programming and Agile Processes in Software Engineering, Oulu, Finland.
- Melo, C., Cruzes, D. S., Kon, F., & Conradi, R. (2011, August 8-12). *Agile Team Perceptions of Productivity Factors*. Paper presented at the AGILE Conference, Salt Lake City, UT.
- Moazeni, R., Link, D., & Boehm, B. (2013). *Incremental development productivity decline*. Paper presented at the 9th International Conference on Predictive Models in Software Engineering, Baltimore, Maryland.
- Mobley, W. H. (1977). Intermediate Linkages in the Relationship Between Job Satisfaction and Employee Turnover. *Journal of Applied Psychology* 62(2), 237-240.

- Moe, N. B., & Šmite, D. (2008). Understanding a lack of trust in Global Software Teams: a multiple-case study. *Software Process: Improvement and Practice*, 13(3), 217-231. doi: 10.1002/spip.378
- Mohagheghi, P., & Conradi, R. (2007). Quality, productivity and economic benefits of software reuse: a review of industrial studies. *Empirical Software Engineering*, 12(5), 471-516.
- Morales, A. W. (2005). Salary survey 2005. *Software Development*, 13(11), 32-42.
- Morrow, P. C., & McElroy, J. C. (1987). Work commitment and job satisfaction over three career stages. *Journal of Vocational Behavior*, 30(3), 330-346. doi: 10.1016/0001-8791(87)90009-1
- Mundel, M. E. (1975). *Measuring and Enhancing the Productivity of Service and Government Organizations*. Tokyo: Asian Productivity Organization.
- Myers, G. J. (1975). *Reliable software through composite design*: Petrocelli/Charter.
- Myrtveit, I., & Stensrud, E. (1999). *Benchmarking COTS Projects Using Data Envelopment Analysis*. Paper presented at the 6th International Symposium on Software Metrics (METRICS '99), Boca Raton, Florida.
- Nachum, L. (1999). Measurement of productivity of professional services. *International Journal of Operations and Production Management*, 9(9/10), 922-950.
- Niederman, F., & Sumner, M. (2003). *Decision paths affecting turnover among information technology professionals*. Paper presented at the Computer Personnel Research (SIGMIS CPR '03).
- Niederman, F., & Sumner, M. R. (2001). *Job turnover among MIS professionals: an exploratory study of employee turnover*. Paper presented at the Computer Personnel Research Conference (SIGCPR'01), San Diego, CA.
- Otero, L. D., Centeno, G., Otero, C. E., & Reeves, K. (2012). A DEA–Tobit Analysis to Understand the Role of Experience and Task Factors in the Efficiency of Software Engineers. *IEEE Transactions on Engineering Management*, 59(3), 391-400. doi: 10.1109/TEM.2010.2101078
- Paiva, E., Barbosa, D., Lima, R., & Albuquerque, A. (2010). Factors that Influence the Productivity of Software Developers in a Developer View. In T. Sobh & K. Elleithy (Eds.), *Innovations in Computing Sciences and Software Engineering* (pp. 99-104): Springer Netherlands.

- Paradi, J. C., Smith, S., & Schaffnit-Chatterjee, C. (2002). Knowledge worker performance analysis using DEA: an application to engineering design teams at Bell Canada. *IEEE Transactions on Engineering Management*, 49(2), 161-172. doi: 10.1109/TEM.2002.1010884
- Patton, M. Q. (1980). *Qualitative evaluation methods*. Beverly Hills, LA: Sage.
- Pendharkar, P. C. (2006). Scale economies and production function estimation for object-oriented software component and source code documentation size. *European Journal of Operational Research*, 172(3), 1040-1050. doi: 10.1016/j.ejor.2004.10.023
- Pendharkar, P. C., & Rodger, J. A. (2009). The relationship between software development team size and software development cost. *Communications of the ACM*, 52(1), 141-144. doi: 10.1145/1435417.1435449
- Peters, L. (2003). *Managing software professionals*. Paper presented at the Engineering Management Conference (IEMC'03).
- Petersen, K. (2011). Measuring and predicting software productivity: A systematic map and review. *Information and Software Technology*, 53(4), 317-343. doi: 10.1016/j.infsof.2010.12.001
- Petticrew, M., & Roberts, H. (2005). *Systematic Reviews in the Social Sciences: A Practical Guide*. Malden, MA: Wiley-Blackwell.
- Pfleeger, S. L. (2008). Software Metrics: Progress after 25 Years? *IEEE Software*, 25(6), 32-34. doi: 10.1109/ms.2008.160
- Phipps, G. (1999). Comparing observed bug and productivity rates for Java and C++. *Software - Practice & Experience*, 29(4), 345-358.
- Premraj, R., Shepperd, M., Kitchenham, B. A., & Forselius, P. (2005). *An Empirical Analysis of Software Productivity over Time*. Paper presented at the Proceedings of the 11th IEEE International Software Metrics Symposium.
- Procaccino, J. D., Verner, J. M., Shelfer, K. M., & Gefen, D. (2005). What do software practitioners really think about project success: an exploratory study. *Journal of Systems and Software*, 78(2), 194-203. doi: 10.1016/j.jss.2004.12.011
- Purao, S., & Vaishnavi, V. (2003). Product metrics for object-oriented systems. *ACM Computing Surveys*, 35(2), 191-221. doi: 10.1145/857076.857090

- Quesnay, F. (1766). Analyse de la formule arithmétique du tableau économique de la distribution des dépenses annuelles d'une nation agricole. *Journal de l'Agriculture, du Commerce & des Finances*, 11-41.
- Ramirez, Y. W., & Nembhard, D. A. (2004). Measuring knowledge worker productivity: A taxonomy. *Journal of Intellectual Capital*, 5(4), 602-628.
- Rasch, R. H. (1991). *An investigation of factors that impact behavioral outcomes of software engineers*. Paper presented at the Computer Personnel Research Conference (SIGCPR'91), Athens, GA.
- Rasch, R. H., & Tosi, H. L. (1992). Factors affecting software developers' performance: an integrated approach. *MIS Quarterly*, 16(3), 395-413. doi: 10.2307/249535
- Rees, A. (1980). Improving Productivity Measurement. *The American Economic Review*, 70(2), 340-342.
- Richens, E. (1998, March 26-28). *HR strategies for IS professionals in the 21st century*. Paper presented at the Computer Personnel Research Conference (SIGCPR'98), Boston, MA.
- Ridings, C. M., & Eder, L. B. (1999). An Analysis of IS technical career paths and job satisfaction. *SIGCPR Computer Personnel*, 20(2), 7-26.
- Rodríguez, D., Sicilia, M. A., García, E., & Harrison, R. (2012). Empirical findings on team size and productivity in software development. *Journal of Systems and Software*, 85(3), 562-570. doi: 10.1016/j.jss.2011.09.009
- Rombach, H. D. (1990). Design measurement: some lessons learned. *IEEE Software*, 7(2), 17-25. doi: 10.1109/52.50770
- Roznowski, M., & Hulin, C. (1992). The scientific merit of valid measures of general constructs with special reference to job satisfaction and job withdrawal. In C. J. Cranny, P. C. Smith & E. F. Stone (Eds.), *Job satisfaction* (pp. 123-163). New York: Lexington Books.
- Ruan, L., Wang, Y., Wang, Q., Li, M., Yang, Y., Xie, L., . . . Dai, J. (2007). *Empirical study on benchmarking software development tasks*. Paper presented at the International Conference on Software Process, Minneapolis, MN.
- Rubin, H., & Rubin, I. (2011). *Qualitative Interviewing: The Art of Hearing Data* (3 ed.). Thousand Oaks, CA: Sage Publications.

- Rubin, H. I., & Hernandez, E. F. (1988). *Motivations and behaviors of software professionals*. Paper presented at the Computer Personnel Research Conference (SIGCPR'88), College park, Maryland.
- Rus, I., & Lindvall, M. (2002). Knowledge management in software engineering. *IEEE Software*, 19(3), 26-38.
- Saari, L. M., & Judge, T. A. (2004). Employee attitudes and job satisfaction. *Human Resources Management*, 43(4), 395-407.
- Sackman, H., Erikson, W. J., & Grant, E. E. (1968). Exploratory experimental studies comparing online and offline programming performance. *Communications of the ACM*, 11(1), 3-11. doi: 10.1145/362851.362858
- Sach, R., Sharp, H., & Petre, M. (2011). *Software Engineers' Perceptions of Factors in Motivation: The Work, People, Obstacles*. Paper presented at the International Symposium on Empirical Software Engineering and Measurement (ESEM '11), Banff, Canada.
- Santana, M., & Robey, D. (1995). Perceptions of control during systems development: effects on job satisfaction of systems professionals. *SIGCPR Computer Personnel*, 16(1), 20-34.
- Santillo, L., & Moretto, G. (2011). Software Productivity Factors Taxonomy (S. S. B. Committee), Trans.). Madrid: GUFPI-ISMA.
- Sawyer, S. (2004). Software development teams. *Communications of the ACM*, 47(12), 95-99. doi: 10.1145/1035134.1035140
- Scacchi, W. (1994). Understanding Software Productivity. In W. D. Hurley (Ed.), *Software Engineering and Knowledge Engineering: Trends for the Next Decade* (Vol. 3, pp. 293-321). Pittsburgh, PA.
- Scudder, R. A., & Kucic, A. R. (1991). Productivity measures for information systems. *Information & Management*, 20(5), 343-354.
- Schuler, R. S., & Jackson, S. E. (1987). Linking Competitive Strategies with Human Resource Management Practices. *The Academy of Management Executive* (1987-1989), 1(3), 207-219
- Shepperd, J. A. (1993). Productivity Loss in Performance Groups: A Motivation Analysis. [doi:]. *Psychological Bulletin*, 113(1), 67-81.
- Simmons, D. (1991). Communications: a software group productivity dominator. *Software Engineering Journal*, 6(6), 454-462.

- Sink, D. S., Tuttle, T. C., & DeVries, S. J. (1984). Productivity measurement and evaluation: what is available? *National Productivity Review*, 3(3), 265-287.
- Sison, R. (2009, 1-3 Dec. 2009). *Investigating the Effect of Pair Programming and Software Size on Software Quality and Programmer Productivity*. Paper presented at the Asia-Pacific Software Engineering Conference (APSEC '09).
- Smits, S. J., McLean, E. R., & Tanner, J. R. (1992). *Managing high achieving information systems professionals*. Paper presented at the Computer Personnel Research Conference (SIGCPR'92), Cincinnati, Ohio.
- Smits, S. J., McLean, E. R., & Tanner, J. R. (1997). *A longitudinal study of I/S careers: synthesis, conclusion, and recommendations*. Paper presented at the Computer Personnel Research Conference (SIGCPR'97).
- Society, I. C. (2004). *SWEBOK: Software Engineering Body of Knowledge*: EUA: Angela Burgess.
- Sommerville, I. (2010). *Software Engineering* (9th ed.). Reading, MA: Addison-Wesley.
- Spector, P. E. (1985). Measurement of human service staff satisfaction: Development of the Job Satisfaction Survey. *American Journal of Community Psychology*, 13(6), 693-713. doi: 10.1007/BF00929796
- Stensrud, E., & Myrtveit, I. (2003). Identifying High Performance ERP Projects. *IEEE Transactions on Engineering Management*, 29(5), 398-416. doi: 10.1109/TSE.2003.1199070
- Strauss, A., & Corbin, J. (1990). *Basics of qualitative research*. Newbury Park, CA: Sage.
- Sudhakar, G. P., Farooq, A., & Patnaik, S. (2011). Soft factors affecting the performance of software development teams. *Team Performance Management* 17(3/4), 187-205. doi: 10.1108/13527591111143718
- Tan, T., Li, Q., Boehm, B. W., Yang, Y., He, M., & Moazeni, R. (2009). *Productivity trends in incremental and iterative software development*. Paper presented at the Proceedings of the 2009 3rd International Symposium on Empirical Software Engineering and Measurement.
- Tangen, S. (2005). Demystifying productivity and performance. *International Journal of Productivity and Performance Management*, 54(1), 34-46.

- Tanner, F. R. (2003, November 2-4). *On motivating engineers*. Paper presented at the Engineering Management Conference (IEMC'03).
- Tausworthe, R. C. (1982). Staffing Implications of Software Productivity Models. *TDA Progress Report* (pp. 42-72).
- Taylor, F. W. (1911). *The principles of scientific management*.
- Tessem, B., & Maurer, F. (2007, June 18-22). *Job Satisfaction and Motivation in a Large Agile Team*. Paper presented at the 8th international conference on Agile processes in software engineering and extreme programming (XP'07), Como, Italy.
- Thadhani, A. J. (1984). Factors affecting programmer productivity during application development. *IBM Systems Journal*, 23(1), 19-35. doi: 10.1147/sj.231.0019
- Thatcher, J. B., Liu, Y., & Stepina, L. P. (2002, May 13-15). *The role of the work itself: An empirical examination of intrinsic motivation's influence on IT workers attitudes and intentions*. Paper presented at the Computer Personnel Research Conference (SIGCPR'02), Kristiansand, Norway.
- Trendowicz, A., & Münch, J. (2009). Factors Influencing Software Development Productivity - State of the Art and Industrial Experiences. In V. Z. Marvin (Ed.), *Advances in Computers* (Vol. 77, pp. 185-241): Elsevier.
- Tsunoda, M., Monden, A., Yadohisa, H., Kikuchi, N., & Matsumoto, K. (2009). Software development productivity of Japanese enterprise applications. *Information Technology and Management*, 10(4), 193-205. doi: 10.1007/s10799-009-0050-9
- Turcotte, J., & Rennison, L. W. (2004). The link between technology use, human capital, productivity and wages: Firm-level evidence. *International Productivity Monitor*, 9(3), 25-36.
- Vessey, I., & Weber, R. (1983). Some factors affecting program repair maintenance: an empirical study. *Communications of the ACM*, 26(2), 128-134. doi: 10.1145/358024.358057
- Vosburgh, J., Curtis, B., Wolverton, R., Albert, B., Malec, H., & Hoben, S. (1984, March 26-29). *Productivity factors and programming environments*. Paper presented at the 7th International Conference on Software Engineering (ICSE'84), Orlando, FL.
- Vroom, V. H. (1964). *Work and motivation*. Oxford, England Wiley.

- Wagner, S., & Ruhe, M. (2008a). A structured review of productivity factors in software development: TU München.
- Wagner, S., & Ruhe, M. (2008b, December). *A Systematic Review of Productivity Factors in Software Development*. Paper presented at the 2nd International Workshop on Software Productivity Analysis and Cost Estimation (SPACE 2008), Beijing (China).
- Walston, C. E., & Felix, C. P. (1977). A method of programming measurement and estimation. *IBM Systems Journal*, 16(1), 54-73. doi: 10.1147/sj.161.0054
- Warr, P., Cook, J., & Wall, T. (1979). Scales for the measurement of some work attitudes and aspects of psychological well-being. *Journal of Occupational Psychology*, 52(2), 129-148. doi: 10.1111/j.2044-8325.1979.tb00448.x
- Weng, W., Su, J., Chen, G., & Wang, Z. (2010, August 24-26). *An Approach for Allocation Optimization of Multi-Project Human Resource Based on DEA*. Paper presented at the International Conference on Management and Service Science (MASS'10), Wuhan, China.
- Westlund, S. G., & Hannon, J. C. (2008). Retaining talent: Assessing job satisfaction facets most significantly related to software developer turnover intentions. *Journal of Information Technology Management*, 19(4), 1-15.
- Wohlin, C., & Ahlgren, M. (1995). Soft factors and their impact on time to market. *Software Quality Journal*, 4(3), 189-205.
- Yang, Z., & Paradi, J. C. (2004). DEA evaluation of a Y2K software retrofit program. *IEEE Transactions on Engineering Management*, 51(3), 279-287. doi: 10.1109/tem.2004.830843
- Yang, Z., & Paradi, J. C. (2009, December 8-11). *A DEA Evaluation of Software Project Efficiency*. Paper presented at the IEEE International Conference on Industrial Engineering and Engineering Management (IEEM 2009), Hong Kong.
- Ye, Y., & Kishida, K. (2003). *Toward an understanding of the motivation Open Source Software developers*. Paper presented at the 25th International Conference on Software Engineering (ICSE'03), Portland, Oregon.
- Yilmaz, M., & O'Connor, R. V. (2012). Social Capital as a Determinant Factor of Software Development Productivity: An Empirical Study Using Structural Equation Modeling. *International Journal of Human Capital and Information Technology Professionals*, 3(2), 40-62. doi: 10.4018/jhicitp.2012040104

- Yilmaz, M., O'Connor, R. V., & Clarke, P. (2012). A Systematic Approach to the Comparison of Roles in the Software Development Processes. In A. Mas, A. Mesquida, T. Rout, R. V. O'Connor & A. Dorling (Eds.), *Software Process Improvement and Capability Determination* (Vol. 290, pp. 198-209): Springer Berlin Heidelberg.
- Yu, B. (2010, August 24-26). *The research of software project performance evaluation model based-on DEA*. Paper presented at the International Conference on Management and Service Science (MASS'10), Wuhan, China.
- Yu, W. D., Smith, D. P., & Huang, S. T. (1991, September 11-13). *Software productivity measurements* Paper presented at the 15th Annual International Computer Software and Applications Conference (COMPSAC'91), Tokyo, Japan.
- Zawacki, R. A. (1992). Motivating the IS people of the future. *Information systems management*, 9(2), 73-75. doi: 10.1080/10580539208906871
- Zhang, S., Wang, Y., Yang, Y., & Xiao, J. (2008). *Capability Assessment of Individual Software Development Processes Using Software Repositories and DEA*. Paper presented at the International Conference on Software Process (ICSP'08).

Anexos

Anexo A - Protocolo revisión sistemática

A continuación se presenta la versión final del protocolo de revisión sistemática de la literatura creado para la presente investigación.

Justificación

Por un lado, la información existente en la literatura científica acerca de la medición de la productividad en ingeniería del software es algo dispersa. A nivel macro, son numerosos los estudios que tratan la medición de la productividad del proceso productivo como un todo, o a nivel organizacional, centrándose principalmente en las líneas de código y/o en los puntos de función como salidas y en las horas-hombre como entradas. Pero por el contrario, la medición a nivel micro, a nivel de puestos de trabajo está prácticamente inexplorada, algunos estudios la abordan desde el punto de vista de los objetivos o desde el desempeño. En todos los niveles no hay consenso sobre cuáles son las medidas ideales a utilizar, aunque parece claro que deben basarse en las características propias del objeto de medición para conseguir una medida ajustada a la realidad.

Por otro lado, es importante de cara a la realización de cualquier investigación posterior, en este caso una tesis doctoral, obtener una visión detallada del estado actual de la cuestión antes de continuar con cualquier trabajo específico. Por ello, se plantea una revisión sistemática de los elementos que sirven para medir la productividad en la ingeniería del software; entendiendo la productividad bajo la fórmula:

$$Productividad = ((Salidas * Factores Salidas) / (Entradas * Factores Entradas)) * Factores Globales$$

Con esta revisión, se pretende obtener una visión global de las salidas y entradas utilizadas para medir la productividad en la ingeniería del software. No es objetivo de esta revisión el realizar un posterior meta-análisis. Para la realización de este protocolo y para la posterior revisión sistemática se ha utilizado como fuente de referencia el informe técnico elaborado por Kitchenham (2007).

Preguntas

P.1. ¿Cuáles son las entradas y salidas del proceso productivo interno de la ingeniería del software?

P.2. ¿Son distintas las entradas y salidas para los diversos puestos de trabajo dentro de la ingeniería del software?

Si la respuesta es afirmativa, entonces se plantan las siguientes preguntas:

P.2.1. ¿Cuáles son las entradas y salidas para el ingeniero de software²⁴?

P.2.2. ¿Cuáles son las entradas y salidas para el programador²⁵?

P.2.3. ¿Cuáles son las entradas y salidas para el analista²⁶?

P.2.4. ¿Cuáles son las entradas y salidas para el jefe de proyecto²⁷?

²⁴ <http://www.onetonline.org/link/summary/15-1132.00>

²⁵ <http://www.onetonline.org/link/summary/15-1131.00>

²⁶ <http://www.onetonline.org/link/summary/15-1121.00>

²⁷ <http://www.onetonline.org/link/summary/11-3021.00>

Protocolo

Criterios de Selección de Fuentes

Disponibilidad para consultar los artículos en la web a través de los acuerdos de la UC3M y la UAM con diversas revistas y publicaciones publicadas desde 1990. A estos artículos es posible acceder mediante usuario y contraseña, o utilizando un ordenador dentro de la red propia de dichas universidades.

Idiomas

Sólo serán analizados los estudios cuyo contenido esté íntegramente escrito en inglés, ya que se considera que la literatura existente en este idioma es la más importante.

Estrategia de Búsqueda

No hay una estrategia de búsqueda publicada previamente para realizar una revisión sistemática entorno a la medición de la productividad en ingeniería del software. Los autores de la presente investigación sólo conocen dos revisiones sistemáticas existentes relativas a la productividad de la ingeniería del software, pero su objetivo es revisar los factores que afectan a la productividad (Paiva, Barbosa, Lima & Albuquerque, 2010; Wagner & Ruhe, 2008). Por otro lado, no hemos encontrado una revisión sistemática entorno a la medición de la productividad en trabajos con fuerte carga de conocimiento y humana, aunque existen otros trabajo exploratorios sin metodología, como por ejemplo una taxonomía de medición de productividad (Ramirez & Nemhard, 2004). Así pues, la estrategia de búsqueda, que a continuación se presenta, se ha diseñado desde cero para el presente estudio.

Lista de fuentes de datos

Las fuentes de datos a utilizar son:

- IEEEExplore.
- EI Compendex.
- ISI Web of Science (WoS).
- ACM Digital Library.

A continuación se incluyen las fuentes de datos relativas a conferencias que serán utilizadas en la búsqueda, no serán utilizadas otras conferencias en la búsqueda.

- International Software Metrics Symposium (METRICS, 1993-2005).
- International Symposium on Empirical Software Engineering (ISESE, 2002-2006).
- International Symposium on Empirical Software Engineering and Measurement (ESEM, 2007-Actualmente).
- International Conference on Software Engineering (ICSE, 1988-Actualmente).

Palabras clave y sinónimos

Lista de palabras clave para las búsquedas y sus sinónimos.

- Productividad: rendimiento.
- Entrada: recurso.
- Salida: producto, servicio.
- Programador.
- Jefe de Proyecto.
- Analista.
- Ingeniero Software.
- Ingeniería del Software: desarrollo de software, mantenimiento de software.

Hay que destacar que los sinónimos de ingeniería del software son dos grandes áreas dentro de la ingeniería que representan gran parte de la actividad y son en las que los puestos de trabajo bajo estudio tienen influencia directa.

Traducción de las palabras claves al idioma de búsqueda:

- Productivity: performance.
- Input: resource.
- Output: product, service.
- Computer Programmer: programmer.
- Project Manager.
- Analyst.
- Software Engineer.
- Software Engineering: software development, software maintenance.

Población de estudio

La población objeto de estudio son las entradas (p. ej. horas-hombre, conocimiento...) y salidas (p. ej. SLOC, PF, requisitos, diseños, documentación...) de los puestos de trabajo relativos al proceso productivo de la ingeniería del software (p. ej. ingeniero software, programador, jefe de proyecto, analista, diseñador...).

Aplicación

Los resultados de la revisión sistemática serán de interés tanto para los investigadores del área de ingeniería del software, en concreto, para aquellos interesados en la medición de la productividad y elementos relacionados tales como la predicción de proyectos software o la medición de beneficios. Y por otro lado, para los gestores y directores de proyectos y los responsables de recursos humanos, ya que dispondrán de más conocimiento sobre los elementos que entran, se transforman y salen durante el proceso productivo de la ingeniería del software.

Selección de estudios primarios

Los resultados serán ordenados por:

- Número de artículos por año por fuente.
- Número de artículos candidatos por año por fuente.
- Número de artículos seleccionados por año por fuente.

Los estudios relevantes serán seleccionados por un solo investigador. Los estudios rechazados serán revisados por otro investigador. Se mantendrá una lista de los artículos candidatos que sean rechazados junto con la razón para su rechazo.

Proceso de búsqueda

En primer lugar se realizará una búsqueda exploratoria con diversas cadenas utilizando las palabras clave y operadores lógicos AND y OR:

("software engineering" OR "software development" OR "software maintenance") AND (productivity OR performance) AND (input OR resource OR output OR product OR service) AND ("computer programmer" OR "programmer" OR "project manager" OR analyst OR "software engineer")

El año de inicio será 1981 dado que a partir de ese año se empiezan a publicar los trabajos más referenciados sobre mediciones de esfuerzos y tamaños en ingeniería del software (Albrecht & Gaffney, 1983; Boehm, 1981; DeMarco, 1986). Estos trabajos crearon un marco de trabajo en los años posteriores para los trabajos en todas las áreas potencialmente relacionadas, entre ellas, la productividad.

Una vez recuperados los artículos se realizará un procedimiento de revisión en el que se aplicarán los siguientes criterios de inclusión y exclusión:

- Criterios de inclusión
 - Artículos que tengan contenido empírico.
 - Artículos sobre medición de la productividad en la ingeniería del software.
- Criterios de exclusión
 - Artículos que no tengan contenido empírico.
 - Artículos que incluyan sólo factores y no tratan sobre entradas o salidas.
 - Artículos que repitan el uso de una misma muestra.

- Artículos que no estén relacionados con la ingeniería del software.
- Artículos que midan la productividad a niveles superiores (p. ej. grupo de trabajo, equipo, departamento, organización...) y no tengan en cuenta el nivel individual.

El procedimiento para seleccionar estudios será el siguiente. En primer lugar, se ejecutará la cadena de búsqueda en los distintos motores de búsqueda. En segundo lugar, para seleccionar un conjunto inicial de estudios, los resúmenes de todos los estudios recuperados serán leídos y evaluado de acuerdo con los criterios de inclusión y exclusión. En tercer lugar, para afinar el conjunto inicial de estudios, cada artículo entero será recuperado y leído para verificar su inclusión o exclusión. El motivo de exclusión o inclusión en esta tercera etapa será documentado, mientras que para la etapa anterior no se realizará esta documentación. Una vez seleccionados estos estudios primarios, sus referencias serán analizadas para identificar posibles estudios que no hayan sido encontrados en la búsqueda y que deban ser analizados en esta revisión sistemática. Este procedimiento (ver Ilustración 26) está basado en el utilizado en una revisión sistemática reciente (Afzal, Torkar & Feldt, 2009).

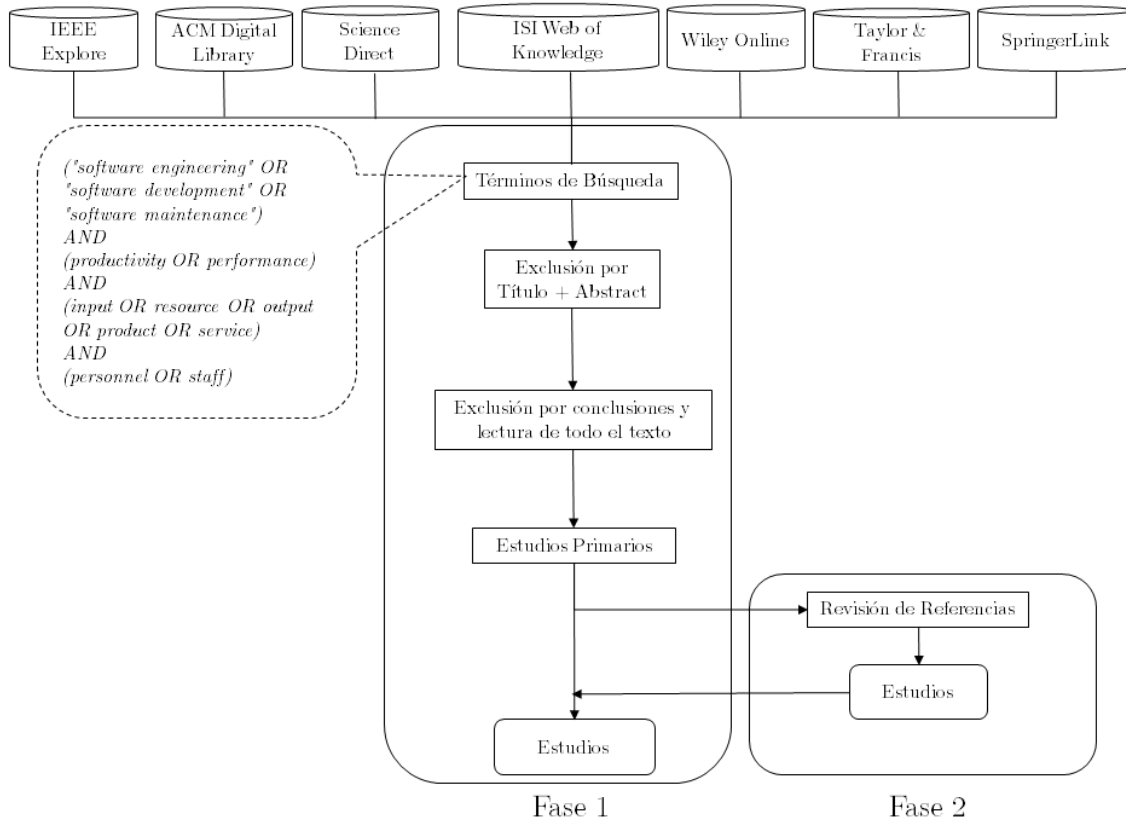


Ilustración 26. Proceso de filtrado; basado en (Afzal, Torkar & Feldt, 2009)

El almacenamiento de los estudios, tanto de la información como de los artículos enteros, se realizará en *endNote X4*. En este programa se crearán dos grupos, uno para los incluidos y otro para los excluidos, además de otras agrupaciones necesarias para realizar el trabajo tales como las agrupaciones de las fases de selección de trabajo. Además, se creará un tipo de referencia específico, denominado *Systematic Review*, con los campos que sean necesarios para rellenar la ficha de cada una de las referencias encontradas.

Extracción de datos

Criterios de inclusión y exclusión del contenido

El estudio debe contener información sobre entradas y/o salidas del proceso productivo en ingeniería del software a nivel interno, o información sobre las entradas

y/o salidas de cada uno de los puestos de interés del estudio, o ha de presentar una fórmula de medición de productividad aplicada a la ingeniería del software.

Formulario de extracción de datos

Cada formulario de extracción de datos contiene datos generales relativos al elemento y su publicación, extraídos de la información almacenada en el gestor de referencias. Por otro lado, contiene datos específicos que son los que posteriormente serán analizados. Entre los datos a analizar destacan los datos de interés para la investigación: las entradas y salidas utilizadas, y la fórmula de productividad empleada, junto con datos de valoración del estudio: tipo y tamaño de fuente de datos.

Con estos datos será posible obtener información cualitativa y cuantitativa para dar respuesta a las preguntas de información planteadas. En concreto, se plantea la obtención de los siguientes tipos de información:

- Tipología de entradas.
- Tipos de entradas por puestos de trabajo.
- Tipología de salidas.
- Tipología de salidas por puestos de trabajo.
- Fórmula de productividad utilizada.
- Fórmula de productividad por puestos de trabajo.

Datos generales	Nombre de la revisión	
	Fecha de la extracción	
	Título	
	Autores	
	Fuente (Revista/Conferencia)	
	Detalles de publicación	
	Resumen del artículo	
Datos	Entradas utilizadas	

específicos	Salidas utilizadas	
	Formula de productividad utilizada	
	Objeto de medición (puesto, proceso, ...)	
	Calidad del estudio	
	Tipo de fuente de datos	
	Tamaño de fuente de datos	
	Otra información	

Medición de las salidas

Las medidas que se utilizarán sobre los resultados se centran en:

- Tipología de entradas
 - Tipos de entradas por puestos de trabajo.
- Tipología de salidas
 - Tipología de salidas por puestos de trabajo.
- Fórmula de productividad utilizada
 - Fórmula de productividad por puestos de trabajo.

Análisis de Datos

Los datos serán tabulados (por orden alfabético del primer autor) para mostrar la información básica de cada estudio. Las tablas serán revisadas para dar respuesta a las preguntas de investigación de la siguiente manera:

P.1. ¿Cuáles son las entradas y salidas del proceso productivo interno de la ingeniería del software? Para esta pregunta se revisarán las entradas y salidas anotadas sobre cada artículo en la tabla y se listarán las entradas,

P.2. ¿Son distintas las entradas y salidas para los diversos puestos de trabajo dentro de la ingeniería del software? Para esta pregunta se seleccionarán los artículos que tengan por objeto la medición de la productividad a nivel de puesto de trabajo, se listarán las entradas y salidas por puestos junto con datos estadísticos descriptivos de cada caso.

P.2.1. ¿Cuáles son las entradas y salidas para el ingeniero de software? Basándose en los resultados de la pregunta P. 2. se seleccionarán las entradas y salidas para este puesto de trabajo.

P.2.2. ¿Cuáles son las entradas y salidas para el programador? Basándose en los resultados de la pregunta P. 2. se seleccionarán las entradas y salidas para este puesto de trabajo.

P.2.3. ¿Cuáles son las entradas y salidas para el analista? Basándose los resultados de la pregunta P. 2. se seleccionarán las entradas y salidas para este puesto de trabajo.

P.2.4. ¿Cuáles son las entradas y salidas para el jefe de proyecto? Basándose en los resultados de la pregunta P. 2. se seleccionarán las entradas y salidas para este puesto de trabajo.

Diseño Experimental

No se va a utilizar ningún método estadístico ya que el objetivo de la revisión sistemática es simplemente exploratorio.

Publicación de Resultados

El informe técnico será publicado como trabajo final de Máster en Dirección de Recursos Humanos en la Universidad Autónoma de Madrid. A partir de los resultados se publicará un artículo científico en una revista internacional de impacto.

Bibliografía

Afzal, W., Torkar, R., & Feldt, R. (2009). A systematic review of search-based testing for non-functional system properties. *Information and Software Technology*, 51(6), 957-976.

- Albrecht, A. J., & Gaffney, J. E. (1983). Software Function, Source Lines of Code, and Development Effort Prediction: A Software Science Validation. *IEEE Transaction Software Engineering*, 9(6), 639-648.
- Boehm, B. W. (1981). *Software Engineering Economics*. Upper Saddle River, NJ: Prentice Hall PTR.
- DeMarco, T. (1986). *Controlling Software Projects: Management, Measurement, and Estimates*. Upper Saddle River, NJ: Prentice Hall PTR.
- Kitchenham, B. A. (2007). *Guidelines for performing Systematic Literature Reviews in Software Engineering*: Keele University and University of Durham.
- Paiva, E., Barbosa, D., Lima, R., & Albuquerque, A. (2010). Factors that Influence the Productivity of Software Developers in a Developer View. In T. Sobh & K. Elleithy (Eds.), *Innovations in Computing Sciences and Software Engineering* (pp. 99-104): Springer Netherlands.
- Ramirez, Y. W., & Nembhard, D. A. (2004). Measuring knowledge worker productivity: A taxonomy. *Journal of Intellectual Capital*, 5(4), 602-628.
- Wagner, S., & Ruhe, M. (2008). *A Systematic Review of Productivity Factors in Software Development*. Paper presented at the 2nd International Workshop on Software Productivity Analysis and Cost Estimation (SPACE 2008), Beijing (China).

Anexo B - Listas de resultados de la revisión sistemática

Fase 1. Lista Inicial

199 IEEE Power Engineering Society Summer Meeting. Conference Proceedings (Cat. No.99CH36364). (1999, 18-22 Jul 1999). Paper presented at the IEEE Power Engineering Society Summer Meeting.

Clinical Epilepsy: Pediatrics. (2005). *Epilepsia*, 46, 142-167.

Clinical Neurophysiology: EEG-Video Monitoring. (2005). *Epilepsia*, 46, 23-34.

Human Imaging: Adult. (2005). *Epilepsia*, 46, 35-48.

Non-AED/Non-Surgical Treatments (Hormonal, Ketogenic Diet, VNS, etc.): All Ages. (2005). *Epilepsia*, 46, 226-235.

Nursing/Psychosocial/Health Services. (2005). *Epilepsia*, 46, 14-23.

Surgery: Pediatric. (2005). *Epilepsia*, 46, 247-253.

Monday, December 4, 2006 Poster Session III 7:30 a.m. – 4:30 p.m. (2006). *Epilepsia*, 47, 204-289.

Poster Session Monday July 3, 2006 13:30–15:00 Poster Session 1 Adult Epileptology. (2006). *Epilepsia*, 47, 36-68.

Sunday, December 3, 2006 Poster Session II 7:30 a.m.–4:30 p.m. (2006). *Epilepsia*, 47, 119-204.

Wednesday July 5, 2006 13:30–15:00 Poster Session 3 Paediatric Epileptology. (2006). *Epilepsia*, 47, 169-204.

ABSTRACTS. (2007). *Acta Pædiatrica*, 96, 46-61.

Abstracts of the 5th Joint Meeting of the German, Austrian, and Swiss Sections of the International League Against Epilepsy Basle, May 16–19, 2007. (2007). *Epilepsia*, 48, 1-66.

Poster Session I 1:00 p.m.-6:00 p.m. (2007). *Epilepsia*, 48, 1-118.

Poster Session II 8:00 a.m.-6:00 p.m. (2007). *Epilepsia*, 48, 125-238.

Poster Session III 8:00 a.m.-2:00 p.m. (2007). *Epilepsia*, 48, 248-380.

Poster Sessions. (2007). *Epilepsia*, 48, 57-172.

Poster Session 2 11:00 a.m.-7:30 p.m. Professionals in Epilepsy care. (2008). *Epilepsia*, 49, 177-319.

Poster Session 3 8:00 a.m.-1:30 p.m. Translational Research. (2008). *Epilepsia*, 49, 337-475.

Saturday, December 6, 2008 Poster Session 1 1:00 p.m.-6:00 p.m. Clinical Neurophysiology. (2008). *Epilepsia*, 49, 1-166.

Abstracts from the Special Care Dentistry 21st Annual Meeting can be accessed in the electronic version of this issue at interscience.wiley.com. (2009). *Special Care in Dentistry*, 29(5), e1-e15.

Oral Communications. (2009). *Journal of Cardiovascular Electrophysiology*, 20, S1-S59.

Abdel-Hamid, T. K., Sengupta, K., & Hardebeck, M. J. (1994). The effect of reward structures on allocating shared staff resources among interdependent software projects: an experimental investigation. *IEEE Transactions on Engineering Management*, 41(2), 115-125.

- Abdel-Hamid, T. K., Sengupta, K., & Swett, C. (1999). The impact of goals on software project management: An experimental investigation. *MIS Quarterly*, 23(4), 531-555.
- Andersson, I., & Nilsson, K. (2002, January 7-10). *Improving diffusion practices in a software organization*. Paper presented at the 35th Annual Hawaii International Conference on System Sciences, Hawaii, HI.
- Anwar, S., Ramzan, M., Rauf, A., & Shahid, A. A. (2010, April 21-23). *Software Maintenance Prediction Using Weighted Scenarios: An Architecture Perspective*. Paper presented at the International Conference on Information Science and Applications, Seoul, Korea.
- April, A. (2010, Sept. 29-Oct. 2). *Studying Supply and Demand of Software Maintenance and Evolution Services*. Paper presented at the Seventh International Conference on the Quality of Information and Communications Technology, Porto, Portugal.
- Ballou, D. P., & Tayi, G. K. (1996). A decision aid for the selection and scheduling of software maintenance projects. *IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans*, 26(2), 203-212.
- Barghouti, N. S. (1994, 5-7 Oct.). *Separating process model enactment from process execution in Provence*. Paper presented at the Ninth International Software Process Workshop Airlie, VA.
- Barreto, A., Barros, M. d. O., & Werner, C. M. L. (2008). Staffing a software project: A constraint satisfaction and optimization-based approach. *Computers & Operations Research*, 35(10), 3073-3089.
- Bartram, D. (1987). The Development of an Automated Testing System for Pilot Selection: The MICROPAT Project1. *Applied Psychology*, 36(3-4), 279-298.
- Basili, V. R., & Boehm, B. (2001). COTS-based systems top 10 list. *Computer*, 34(5), 91-95.

- Ben-Bassat, M. (1999, 30 Aug.-2 Sept.). *Creating service expertise from raw data with expert system software*. Paper presented at the IEEE Systems Readiness Technology Conference, San Antonio, TX.
- Bertolino, A., Marchetti, E., & Mirandola, R. (2007). Performance measures for supporting project manager decisions. *Software Process: Improvement and Practice*, 12(2), 141-164.
- Bianchi, A., Lanubile, F., & Visaggio, G. (2001, April 4-6). *A controlled experiment to assess the effectiveness of inspection meetings*. Paper presented at the Seventh International Software Metrics Symposium, London, UK.
- Biffl, S., & Halling, M. (2003). Investigating the defect detection effectiveness and cost benefit of nominal inspection teams. *IEEE Transactions on Software Engineering*, 29(5), 385-397.
- Blais, C. L. (1995, Dec. 3-6). *Scalability issues in enhancement of the MAGTF Tactical Warfare Simulation system*. Paper presented at the Simulation Conference, Arlington, VA.
- Boehm, B. W., & Papaccio, P. N. (1988). Understanding and controlling software costs. *IEEE Transactions on Software Engineering*, 14(10), 1462-1477.
- Bommer, M., & Pease, V. (1991). Mitigating the impact of project cancellations on productivity. *National Productivity Review*, 10(4), 453-463.
- Borjesson, A., & Mathiassen, L. (2004). Successful process implementation. *IEEE Software*, 21(4), 36-44.
- Bowski, L., Perley, C. R., & West, J. M. (1983). A minicomputer system for analyzing and reporting pilot plant fermentor data. *Biotechnology and Bioengineering*, 25(5), 1237-1250.

- Butler, T. (2008). Teaching and Learning Guide for: Memoryscape: How Audio Walks Can Deepen Our Sense of Place by Integrating Art, Oral History and Cultural Geography. *Geography Compass*, 2(5), 1750-1754.
- Canfora, G., Cimitile, A., Garcia, F., Piattini, M., & Visaggio, C. A. (2006, March 22-24). *Performances of pair designing on software evolution: a controlled experiment*. Paper presented at the 10th European Conference on Software Maintenance and Reengineering, Bari, Italy.
- Cao, L., Ramesh, B., & Abdel-Hamid, T. (2010). Modeling dynamics in agile software development. *ACM Transactions on Information Systems*, 1(1), 1-26.
- Catal, C., & Diri, B. (2007, February 13-15). *Software defect prediction using artificial immune recognition system*. Paper presented at the IASTED International Conference on Software Engineering, Innsbruck, Austria.
- Chan, T. Z. (2000, Oct. 14). *Beyond productivity in software maintenance: Factors affecting lead time in servicing users' requests*. Paper presented at the International Conference on Software Maintenance, San Jose, CA.
- Chang, K.-c., Sheu, T. S., Klein, G., & Jiang, J. J. (2010). User commitment and collaboration: Motivational antecedents and project performance. *Information and Software Technology*, 52(6), 672-679.
- Chi, D. H., & Kuo, W. (1990). Optimal design for software reliability and development cost. *IEEE Journal on Selected Areas in Communications*, 8(2), 276-282.
- Chi, D. H., Lin, H. H., & Kuo, W. (1989, Jan. 24-26). *Software reliability and redundancy optimization*. Paper presented at the Annual Reliability and Maintainability Symposium, Atlanta, GA.
- Christian, J. T., & Eward, M. M. (1994, Oct. 11-13). *Transferring Software Engineering Technology: The Software Productivity Consortium Experience*. Paper presented at the IFIP TC8 Working Conference on Diffusion, Transfer, and Implementation of Information Technology, Pittsburgh, PA.

- Clark, B. K. (2000). Quantifying the effects of process improvement on effort. *IEEE Software*, 17(6), 65-70.
- Cunningham, T., Medlock, A., Sandefur, M., Rowse, J., & Asme. (2005, June 6-9). *High-productivity, low-cost, gas turbine system modeling software*. Paper presented at the ASME Turbo Expo, Reno-Tahoe, NV.
- Curtis, B., Hefley, W. E., Miller, S., & Konrad, M. (1997). Developing organizational competence. *Computer*, 30(3), 122-124.
- Cusick, J. J., & Ma, G. (2010, April 19-23). *Creating an ITIL inspired Incident Management approach: Roots, response, and results*. Paper presented at the IEEE/IFIP Network Operations and Management Symposium Workshops, Osaka, Japan.
- de Silva Garza, A. G., Franzoni Velázquez, A. L., & Morales, V. C. (2007). Promoting productivity in manufacturing companies in developing countries: An information system for managing and querying knowledge bases in the automotive industry in Mexico. *Information Technology for Development*, 13(3), 253-268.
- Di Penta, M., Casazza, G., Antoniol, G., & Merlo, E. (2001, March 14-16). *Modeling Web maintenance centers through queue models*. Paper presented at the Fifth European Conference on Software Maintenance and Reengineering, Lisbon , Portugal.
- Diamant, E. I., Fussell, S. R., & Lo, F.-l. (2008). *Where did we turn wrong?: unpacking the effect of culture and technology on attributions of team performance*. Paper presented at the ACM conference on Computer supported cooperative work, San Diego, CA.
- Dong, F., Li, M., Li, J., Yang, Y., & Wang, Q. (2009, Oct. 15-16). *Effect of staffing pattern on software project: An empirical analysis*. Paper presented at the 3rd International Symposium on Empirical Software Engineering and Measurement, Lake Buena Vista, FL.
- Dura, O., & Yilmaz, A. E. (2009, Sept. 14-16). *Software product line development: A review on practical issues and challenges*. Paper presented at the 24th International Symposium on Computer and Information Sciences, North Cyprus, Turkey.

- Eilam, T., Kalantar, M. H., Konstantinou, A. V., Pacifici, G., Pershing, J., & Agrawal, A. (2006). Managing the configuration complexity of distributed applications in Internet data centers. *Communications Magazine, IEEE*, 44(3), 166-177.
- El Emam, K., & Birk, A. (2000). Validating the ISO/IEC 15504 measure of software requirements analysis process capability. *IEEE Transactions on Software Engineering*, 26(6), 541-566.
- Erdogmus, H. (2007, Sept. 20-21). *A cost effectiveness indicator for software development*. Paper presented at the First International Symposium on Empirical Software Engineering and Measurement, Madrid, Spain.
- Ericson, C. A. (2011). *System Safety Terms and Concepts*: John Wiley & Sons, Inc.
- Erlikh, L. (2000). Leveraging legacy system dollars for e-business. *IT Professional*, 2(3), 17-23.
- Fægri, T. E., Dyba, T., & Dingsoyr, T. (2010). Introducing knowledge redundancy practice in software development: Experiences with job rotation in support work. *Information and Software Technology*, 52(10), 1118-1132.
- Fanberg, V. (2001, Oct. 8-10). *Use of binary file comparison tools in software release management*. Paper presented at the Second Asia-Pacific Conference on Quality Software, Hong Kong, China.
- Fazio, M. E. (2006, April 1-2). *Design and Development of a Model to Optimize the Clinical Engineering Department Size within a Hospital*. Paper presented at the IEEE 32nd Annual Northeast Bioengineering Conference, Easton, PA.
- Foley, D., & Calliss, F. W. (1993, Sept. 27-30). *Improving the quality of three products through improved testing: A case study*. Paper presented at the Conference on Software Maintenance, Montréal, Canada.
- Frangos, S. A. (1998). Motivated humans for reliable software products. *Microprocessors and Microsystems*, 21(10), 605-610.

- Fullerton, N. (1998, May 21). *Energy Management System architecture*. Paper presented at the IEE Colloquium on The New NIE Energy Management System, Belfast , UK.
- Furth, R. (1996, April 21-24). *A model for professional training and education within a software engineering organization*. Paper presented at the Ninth Conference on Software Engineering Education, Daytona Beach, FL.
- Goebel, C. J. (2009, Aug. 24-28). *How Being Agile Changed Our Human Resources Policies*. Paper presented at the Agile Conference, Chicago, IL.
- Goh, T. N. (2010). An information management paradigm for statistical experiments. *Quality and Reliability Engineering International*, 26(5), 487-494.
- Gopal, A., & Koka, B. R. (2010). The Role of Contracts on Quality and Returns to Quality in Offshore Software Development Outsourcing. *Decision Sciences*, 41(3), 491-516.
- Guo, J., & Luqi. (2000, March 22-26). *A survey of software reuse repositories*. Paper presented at the Seventh IEEE International Conference and Workshop on the Engineering of Computer Based Systems, Edinburgh, UK.
- Hale, J., Parrish, A., Dixon, B., & Smith, R. K. (2000). Enhancing the Cocomo estimation models. *IEEE Software*, 17(6), 45-49.
- Harr, J., & Denault, G. (2002, April 15-19). *Issues concerning Linux clustering: cluster management and application porting*. Paper presented at the International Parallel and Distributed Processing Symposium, Fort Lauderdale, FL.
- Hartmann, J., & Robson, D. J. (1989, Oct. 16-19). *Revalidation during the software maintenance phase*. Paper presented at the Conference on Software Maintenance, Miami, FL.
- Hartmann, J., & Robson, D. J. (1990, Jan. 2-5). *RETEST-development of a selective revalidation prototype environment for use in software maintenance*. Paper presented at the Twenty-Third Annual Hawaii International Conference on System Sciences, Kailua-Kona, HI.

- Heimbigner, D., & Wolf, A. L. (1996, June 17-19). *Software in the field needs process too*. Paper presented at the 10th International Software Process Workshop, Dijon, France.
- Heiskari, J., & Lehtola, L. (2009, 1-3 Dec. 2009). *Investigating the State of User Involvement in Practice*. Paper presented at the 16th Asia-Pacific Software Engineering Conference, Penang, Malaysia.
- Holtzman De Garcia, L. N., Souza Tavares, S. R., & Segre, L. (1993, Dec. 17-18). *Participatory management: a case study in a high tech Brazilian plant*. Paper presented at the IEEE International Engineering Management Conference, New Delhi , India.
- Irani, Z., & Sharp, J. M. (1997, 27-31 Jul 1997). *Re-engineering people through leadership and empowerment: a case study*. Paper presented at the Portland International Conference on Management and Technology Innovation in Technology Management, Portland, OR.
- Jackman, M. (1998). Homeopathic remedies for team toxicity. *IEEE Software*, 15(4), 43-45.
- Jai, A., & Sarkar, S. (2005, 03-06 Jan. 2005). *Staffing Software Maintenance and Support Projects*. Paper presented at the 38th Annual Hawaii International Conference on System Sciences, Big Island, HI.
- Jiamthubthugsin, W., & Sutivong, D. (2006, 04-07 Jan. 2006). *Resource Decisions in Software Development Using Risk Assessment Model*. Paper presented at the 39th Annual Hawaii International Conference on System Sciences, Kauai, HI.
- Jiandong, Z. (2009, 11-13 Dec. 2009). *Employee Engagement Investigation in IT Industry*. Paper presented at the International Conference on Computational Intelligence and Software Engineering, Wuhan, China.
- Jiang, X.-r., & Li, S.-c. (2010, 16-18 July 2010). *The antecedents of business-to business relational stability and its impacts on relational performance*. Paper presented at the IEEE International Conference on Software Engineering and Service Sciences, Beijing, China.

- Jones, C. (1994). Globalisation of software supply and demand. *Software Engineering Journal*, 9(6), 235-243.
- Jones, C. (1995). End user programming. *Computer*, 28(9), 68-70.
- Jones, C. (1995). How office space affects programming productivity. *Computer*, 28(1), 76-77.
- Jordan, G., & Segelod, E. (2006). Software innovativeness: outcomes on project performance, knowledge enhancement, and external linkages. *R&D Management*, 36(2), 127-142.
- Kazemzadeh, R. B., & Bashiri, M. (2005, Sept. 11-13, 2005). *Determination of critical factors on employee satisfaction - a case study on a financial institute*. Paper presented at the IEEE International Engineering Management Conference, St. John's, Newfoundland & Labrador, Canada.
- Kazman, R., Klein, M., Barbacci, M., Longstaff, T., Lipson, H., & Carriere, J. (1998, Aug. 10-14). *The architecture tradeoff analysis method*. Paper presented at the IEEE International Conference on Engineering of Complex Computer Systems, Monterey, CA.
- Khoshgoftaar, T. M., Lanning, D. L., & Pandya, A. S. (1993, Nov. 3-6). *A neural network modeling methodology for the detection of high-risk programs*. Paper presented at the Fourth International Symposium on Software Reliability Engineering, Denver, CO.
- Kieburtz, R. B., McKinney, L., Bell, J. M., Hook, J., Kotov, A., Lewis, J., et al. (1996, March 25-29). *A software engineering experiment in software component generation*. Paper presented at the 18th International Conference on Software Engineering, Berlin, Germany
- Killen, C. P. (1997, July 27-31). *The non-technical side of manufacturing change: Australian case studies*. Paper presented at the Portland International Conference on Management and Technology Innovation in Technology Management, Portland, OR.

- Kitsopoulos, S. C. (1994, Oct. 17-19). *From mentoring to executive coaching-a psychological case study*. Paper presented at the IEEE International Engineering Management Conference, Dayton North, OH.
- Knudstrup, J., Haggouchi, K., Peron, M., Quinn, P., Ballester, P., Banse, K., et al. (2002, Aug. 22). *Evolution and adaptation of the VLT data flow system*. Paper presented at the Observatory Operations to Optimize Scientific Return III, Waikoloa, HI.
- Krishnan, M. S., Kriebel, C. H., Kekre, S., & Mukhopadhyay, T. (2000). An empirical analysis of productivity and quality in software products. *Management Science*, 46(6), 745-759.
- Lacy, J. (1983). Preface. *IBM Systems Journal*, 22(3), 168-169.
- Laird, L. M. (2006). The Limitations of Estimation. *IT Professional*, 8(6), 40-45.
- Lasserre, C. M., Acuña, S. T., & Quincoces, V. E. (2001, Nov. 6-8). *Empiric validation of the person to role allocation process*. Paper presented at the XXI Internatinal Conference of the Chilean Computer Science Society, Punta Arenas, Chile.
- Leonhard, C. A., & Davis, J. S. (1995). Job-shop development model: a case study. *IEEE Software*, 12(2), 86-92.
- Lilley, M. M., & Williams, R. G. (1991, Oct. 27-31). *Partner selection for joint venture agreements*. Paper presented at the Technology Management : the New International Language, Portland, OR.
- Lim, W. C. (1994). Effects of reuse on quality, productivity, and economics. *IEEE Software*, 11(5), 23-30.
- Lirov, Y., Prakash, S., & Ravikumar, S. (1991). ALCA — automated local area networks configuration aid. *Expert Systems*, 8(3), 171-182.
- Little, T. (2004). Value creation and capture: a model of the software development process. *IEEE Software*, 21(3), 48-53.

- Lynex, A., & Layzell, P. J. (1997, July 14-18). *Understanding resistance to software reuse*. Paper presented at the Eighth IEEE International Workshop on Software Technology and Engineering Practice, London , UK.
- Mackey, K. (1998). Beyond Dilbert: creating cultures that work [software industry]. *IEEE Software*, 15(1), 48-49.
- Martin, S., & Kidwell, D. K. (2001, Aug. 16-17). *A case study in cluster analysis for intranet organization*. Paper presented at the 2nd International Workshop on Engineering Management for Applied Technology, Austin, TX
- Mauceri, L. J. (1974). Control of an expanding network—"An operational nightmare". *Networks*, 4(4), 287-297.
- McConnell, S. (1998). Problem programmers. *IEEE Software*, 15(2), 126-128.
- McGarry, F., & Decker, B. (2002). Attaining Level 5 in CMM process maturity. *IEEE Software*, 19(6), 87-96.
- Middleton, P., Ho Woo, L., & Irani, S. A. (2004). Why culling software colleagues is popular. *IEEE Software*, 21(5), 28-32.
- Mockus, A., Fielding, R. T., & Herbsleb, J. D. (2002). Two case studies of open source software development: Apache and Mozilla. *ACM Transactions on Software Engineering and Methodology*, 11(3), 309-346.
- Moore, B. J. (1994, Oct. 17-19). *Case study of unforeseen negative impacts of a TQM program*. Paper presented at the IEEE International Engineering Management Conference, Dayton North, OH.
- Nelson, K. M., Nelson, H. J., & Ghods, M. (2001, Jan. 3-6). *Understanding the personal competencies of IS support experts: moving toward the e-business future*. Paper presented at the 34th Annual Hawaii International Conference on System Sciences, Maui, HI.

- Ning, Z., Xiaoting, H., Xiaojun, Z., & Guangyu, X. (2009, Dec. 11-13). *Performance Appraisal System for Academic Staff in the Context of Digital Campus of Higer Education Institutions: Design and Implementation*. Paper presented at the International Conference on Computational Intelligence and Software Engineering, Wuhan, China.
- Ooi, G., & Soh, C. (2003). Developing an activity-based costing approach for system development and implementation. *SIGMIS Database*, 34(3), 54-71.
- Ortiz, E. A. (1995, Jan. 16-19). *Software-quality improvement using reliability-growth models*. Paper presented at the Annual Reliability and Maintainability Symposium, Washington, DC.
- Otero, C. E., Otero, L. D., Weissberger, I., & Qureshi, A. (2010, March 24-26). *A Multi-criteria Decision Making Approach for Resource Allocation in Software Engineering*. Paper presented at the 12th International Conference on Computer Modelling and Simulation, Cambridge, United Kingdom.
- Padberg, F. (2002). A discrete simulation model for assessing software project scheduling policies. *Software Process: Improvement and Practice*, 7(3-4), 127-139.
- Padberg, F. (2002, Dec. 4-6). *Using process simulation to compare scheduling strategies for software projects*. Paper presented at the Ninth Asia-Pacific Software Engineering Conference, Queensland, Australia.
- Padberg, F., & Müller, M. M. (2003, Sept. 3-5). *Analyzing the cost and benefit of pair programming*. Paper presented at the Ninth International Software Metrics Symposium, Sydney, Australia.
- Parkan, C., Lam, K., & Hang, G. (1997). Operational competitiveness analysis on software development. *Journal of the Operational Research Society*, 48(9), 892-905.
- Patanakul, P. (2009, Aug. 2-6). *Toward an understanding of the dynamic of project manager assignments: An empirical study*. Paper presented at the International Conference on Management of Engineering & Technology, Portland, OR.

- Pei, G., & Victoria, A. (1994). Reusability in software maintenance: A Japan—USA comparison. *Journal of Software Maintenance: Research and Practice*, 6(4), 165-183.
- Pfahl, D., Al-Emran, A., & Ruhe, G. (2006). Simulation-based stability analysis for software release plans. In Q. Wang, D. Pfahl, D. M. Raffo & P. Wernick (Eds.), *Software Process Change* (Vol. 3966, pp. 262-273): Springer Berlin / Heidelberg.
- Pfahl, D., Al-Emran, A., & Ruhe, G. (2007). A System Dynamics Simulation Model for Analyzing the Stability of Software Release Plans. *Software Process: Improvement and Practice*, 12(5), 475-490.
- Phister Jr, M. (1981). Computer Capacity Management. *Information & Management*, 4(5), 269-273.
- Phister Jr, M. (1981). A model of the software development process. *Journal of Systems and Software*, 2(3), 237-255.
- Pillai, K., & Nair, V. S. S. (1998, March 26-28). *Early prediction of project schedule slippage*. Paper presented at the IEEE Workshop on Application-Specific Software Engineering Technology, Richardson, TX.
- Prendergast, B. (1995, Nov. 13-15). *Inter-discipline cross-shift teams for station improvements-a case study*. Paper presented at the IEEE Advanced Semiconductor Manufacturing Conference and Workshop, Cambridge, MA.
- Qing, Z., & Zhiming, R. (2010, May 7-9). *Research on Human Resource Configuration Strategy in Software Engineering*. Paper presented at the International Conference on E-Business and E-Government, Guangzhou, China.
- Qingrui, X., Zhangshu, X., & Ling, Z. (2003, Nov. 2-4). *Innovation by everyone: case study from a Chinese top enterprise*. Paper presented at the IEEE International Engineering Management Conference, Estoril, Portugal.

- Ramesh, B., & Edwards, M. (1993, Jan. 4-6). *Issues in the development of a requirements traceability model*. Paper presented at the IEEE International Symposium on Requirements Engineering, San Diego, CA.
- Rauscher, T. G., & Smith, P. G. (1995). From experience time-driven development of software in manufactured goods. *Journal of Product Innovation Management*, 12(3), 186-199.
- Rauscher, T. G., & Smith, P. G. (1995). Time-Driven Development of Software in Manufactured Goods. *Journal of Product Innovation Management*, 12(3), 186-199.
- Rivas, L., Perez, M., Mendoza, L. E., & Griman, A. (2008, Oct. 26-31). *Towards a Selection Model for Software Engineering Tools in Small and Medium Enterprises (SMEs)*. Paper presented at the Third International Conference on Software Engineering Advances, Sliema, Malta.
- Rivera-Ibarra, J. G., Rodriguez-Jacobo, J., & Serrano-Vargas, M. A. (2010, March 9-12). *Competency Framework for Software Engineers*. Paper presented at the 23rd IEEE Conference on Software Engineering Education and Training, Pittsburgh, PA.
- Ropponen, J., & Lyytinen, K. (2000). Components of software development risk: how to address them? A project manager survey. *IEEE Transactions on Software Engineering*, 26(2), 98-112.
- Rothman, J. (1999). Retrain your code czar. *IEEE Software*, 16(2), 86-88.
- Sadr, B., & Dousette, P. J. (1996). An OO project management strategy. *Computer*, 29(9), 33-38.
- Salmeron, J. L., & Lopez, C. (2010). A multicriteria approach for risks assessment in ERP maintenance. *Journal of Systems and Software*, 83(10), 1941-1953.
- Sanders, N. (1996, Jan. 24-27). *Automated testing using executable formal specifications*. Paper presented at the International Conference Software Engineering: Education and Practice, Dunedin , New Zealand.

- Sardos, P. A. (1994, Oct. 17-19). *Quality management-a case study*. Paper presented at the IEEE International Engineering Management Conference, Dayton North, OH.
- Selamat, M. H., Othman, A. T., Rahim, M. M., & Khalil, I. (1993, July 19-23). *A model for CASE implementation: A Malaysian experience*. Paper presented at the Sixth International Workshop on Computer-Aided Software Engineering, Kent Ridge, Singapore.
- Sengupta, K., Abdel-Hamid, T. K., & Bosley, M. (1999). Coping with staffing delays in software project management: an experimental investigation. *IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans*, 29(1), 77-91.
- Shendil, K., & Madhavji, N. H. (1994, Oct. 5-7). *Personal 'progress functions' in the software process*. Paper presented at the Ninth International Software Process Workshop, Airlie, VA.
- Shepperd, M., Schofield, C., & Kitchenham, B. (1996, March 25-29). *Effort estimation using analogy*. Paper presented at the 18th International Conference on Software Engineering, Berlin , Germany.
- Sherer, S. W., Kouchakdjian, A., & Arnold, P. G. (1996). Experience using cleanroom software engineering. *IEEE Software*, 13(3), 69-76.
- Sneed, H. M. (1997, March 17-19). *Measuring the performance of a software maintenance department*. Paper presented at the First Euromicro Conference on Software Maintenance and Reengineering, Berlin , Germany.
- Soedarsono, A. A., Murray, S. L., & Omurtag, Y. (1998). Productivity improvement at a high-tech state-owned industry-an Indonesian case study of employee motivation. *IEEE Transactions on Engineering Management*, 45(4), 388-395.
- Solotorevsky, G., Gudes, E., & Meisels, A. (1991, May 28-29). *Specifying resource allocation and time-tabling problems using a rule-based language*. Paper presented at the Fifth Israel Conference on Computer Systems and Software Engineering, Herzlia, Israel.

- Stark, G. E., & Oman, P. W. (1997). Software maintenance management strategies: observations from the field. *Journal of Software Maintenance: Research and Practice*, 9(6), 365-378.
- Strigel, W. (2002). *Software Productivity Center Inc*: John Wiley & Sons, Inc.
- Swedish, J. A. (1998, Dec. 13-16). *Simulation of an inland waterway barge fleet distribution network*. Paper presented at the Winter Simulation Conference, Washington, DC.
- Tai, A. T., Trivedi, K. S., & Hecht, H. (1997, Aug. 11-12). *On the development of dependability-evaluation workbench for high-assurance system designers*. Paper presented at the High-Assurance Systems Engineering Workshop, 1997., Proceedings.
- Taizan, C. (2000, Oct. 14). *Beyond productivity in software maintenance: factors affecting lead time in servicing users' requests*. Paper presented at the International Conference on Software Maintenance, San Jose, CA.
- Tamrat, E., Vilkinas, T., & Warren, J. R. (1996, Jan. 3-6). *Analysis of a telecommuting experience: a case study*. Paper presented at the 29th Hawaii International Conference on System Sciences, Maui, HI.
- Tan, T., Li, Q., Boehm, B., Yang, Y., He, M., & Moazeni, R. (2009). *Productivity trends in incremental and iterative software development*. Paper presented at the 3rd International Symposium on Empirical Software Engineering and Measurement, Lake Buena Vista, FL.
- Teasley, S., Covi, L., Krishnan, M. S., & Olson, J. S. (2000). *How does radical collocation help a team succeed?* Paper presented at the ACM conference on Computer supported cooperative work, Philadelphia, PA.
- Teng, Y., & He, Z. (2009, May 15-17). *Design and Implementation of Stored Procedure Router Based on Dynamic SQL*. Paper presented at the International Forum on Information Technology and Applications, Chengdu, China.

- Tomaszewski, P., & Lundberg, L. (2006). The increase of productivity over time--an industrial case study. *Information and Software Technology*, 48(9), 915-927.
- Tsagias, M., & Kitchenham, B. (2000). An evaluation of the business object approach to software development. *Journal of Systems and Software*, 52(2-3), 149-156.
- Ureyen, R., Meydanli, I. I., & Koksaldi, S. (1999, July 25-29). *Reorganization of an industrial R&D center: a case study*. Paper presented at the Portland International Conference on Management of Engineering and Technology, Portland, OR.
- Vaishnavi, V. K., Abbott, D., Cooper, T. A., & Walling, K. (1992, Sep. 21-25). *Panel on management of emerging software technologies*. Paper presented at the Sixteenth Annual International Computer Software and Applications Conference, Chicago, IL.
- Velez, R., Zhang, D., & Kho, J. (1993, Nov. 8-11). *An intelligent tool for UNIX performance tuning*. Paper presented at the Fifth International Conference on Tools with Artificial Intelligence, Boston, MA.
- Vucetic, J., & Kline, P. (1998, May 18-20). *Network management applications for wireless local loop*. Paper presented at the 9th Mediterranean Electrotechnical Conference, Tel-Aviv, Israel.
- Wald, B., & Salisbury, L. A. (1977). The Computer Family Architecture Project: Service Perspectives and Overview. *Computer*, 10(10), 8-11.
- Wang, P., & Turban, E. (1994). Management information systems issues of the 1990s in the Republic of China: An industry analysis. *International Journal of Information Management*, 14(1), 25-38.
- Wiegner, R. T., & Nof, S. Y. (1993). The software product feedback flow model for development planning. *Information and Software Technology*, 35(8), 427-438.
- Williams, M., Packlick, J., Bellubbi, R., & Coburn, S. (2007, Aug. 13-17). *How We Made Onsite Customer Work - An Extreme Success Story*. Paper presented at the AGILE, Washington, DC.

Yu, W. D., & Patil, G. (2007, July 1-4). *A Workflow-Based Test Automation Framework for Web Based Systems*. Paper presented at the 12th IEEE Symposium on Computers and Communications, Kerkyra, Greece.

Zeng, Y., Wang, L., & Zhang, J. (2006, May 27-28). *Aggregative risk assessment model for information technology project development*. Paper presented at the Fifth Wuhan International Conference on E-Business, Wuhan, China.

Zhong, X., Madhavji, N. H., & El Emam, K. (2000). Critical factors affecting personal software processes. *IEEE Software*, 17(6), 76-83.

Incluidos

Abdel-Hamid, T. K., Sengupta, K., & Hardebeck, M. J. (1994). The effect of reward structures on allocating shared staff resources among interdependent software projects: an experimental investigation. *IEEE Transactions on Engineering Management*, 41(2), 115-125.

Abdel-Hamid, T. K., Sengupta, K., & Swett, C. (1999). The impact of goals on software project management: An experimental investigation. *MIS Quarterly*, 23(4), 531-555.

Andersson, I., & Nilsson, K. (2002, January 7-10). *Improving diffusion practices in a software organization*. Paper presented at the 35th Annual Hawaii International Conference on System Sciences, Hawaii, HI.

April, A. (2010, Sept. 29-Oct. 2). *Studying Supply and Demand of Software Maintenance and Evolution Services*. Paper presented at the Seventh International Conference on the Quality of Information and Communications Technology, Porto, Portugal.

Barreto, A., Barros, M. d. O., & Werner, C. M. L. (2008). Staffing a software project: A constraint satisfaction and optimization-based approach. *Computers & Operations Research*, 35(10), 3073-3089.

- Ben-Bassat, M. (1999, 30 Aug.-2 Sept.). *Creating service expertise from raw data with expert system software*. Paper presented at the IEEE Systems Readiness Technology Conference, San Antonio, TX.
- Bertolino, A., Marchetti, E., & Mirandola, R. (2007). Performance measures for supporting project manager decisions. *Software Process: Improvement and Practice*, 12(2), 141-164.
- Bianchi, A., Lanubile, F., & Visaggio, G. (2001, April 4-6). *A controlled experiment to assess the effectiveness of inspection meetings*. Paper presented at the Seventh International Software Metrics Symposium, London, UK.
- Biffl, S., & Halling, M. (2003). Investigating the defect detection effectiveness and cost benefit of nominal inspection teams. *IEEE Transactions on Software Engineering*, 29(5), 385-397.
- Boehm, B. W., & Papaccio, P. N. (1988). Understanding and controlling software costs. *IEEE Transactions on Software Engineering*, 14(10), 1462-1477.
- Bommer, M., & Pease, V. (1991). Mitigating the impact of project cancellations on productivity. *National Productivity Review*, 10(4), 453-463.
- Canfora, G., Cimitile, A., Garcia, F., Piattini, M., & Visaggio, C. A. (2006, March 22-24). *Performances of pair designing on software evolution: a controlled experiment*. Paper presented at the 10th European Conference on Software Maintenance and Reengineering, Bari, Italy.
- Cao, L., Ramesh, B., & Abdel-Hamid, T. (2010). Modeling dynamics in agile software development. *ACM Transactions on Information Systems*, 1(1), 1-26.
- Chan, T. Z. (2000, Oct. 14). *Beyond productivity in software maintenance: Factors affecting lead time in servicing users' requests*. Paper presented at the International Conference on Software Maintenance, San Jose, CA.

- Chang, K.-c., Sheu, T. S., Klein, G., & Jiang, J. J. (2010). User commitment and collaboration: Motivational antecedents and project performance. *Information and Software Technology*, 52(6), 672-679.
- Diamant, E. I., Fussell, S. R., & Lo, F.-l. (2008). *Where did we turn wrong?: unpacking the effect of culture and technology on attributions of team performance*. Paper presented at the ACM conference on Computer supported cooperative work, San Diego, CA.
- Dong, F., Li, M., Li, J., Yang, Y., & Wang, Q. (2009, Oct. 15-16). *Effect of staffing pattern on software project: An empirical analysis*. Paper presented at the 3rd International Symposium on Empirical Software Engineering and Measurement, Lake Buena Vista, FL.
- Dura, O., & Yilmaz, A. E. (2009, Sept. 14-16). *Software product line development: A review on practical issues and challenges*. Paper presented at the 24th International Symposium on Computer and Information Sciences, North Cyprus, Turkey.
- Erdogmus, H. (2007, Sept. 20-21). *A cost effectiveness indicator for software development*. Paper presented at the First International Symposium on Empirical Software Engineering and Measurement, Madrid, Spain.
- Fægri, T. E., Dyba, T., & Dingsoyr, T. (2010). Introducing knowledge redundancy practice in software development: Experiences with job rotation in support work. *Information and Software Technology*, 52(10), 1118-1132.
- Frangos, S. A. (1998). Motivated humans for reliable software products. *Microprocessors and Microsystems*, 21(10), 605-610.
- Goebel, C. J. (2009, Aug. 24-28). *How Being Agile Changed Our Human Resources Policies*. Paper presented at the Agile Conference, Chicago, IL.
- Hale, J., Parrish, A., Dixon, B., & Smith, R. K. (2000). Enhancing the Cocomo estimation models. *IEEE Software*, 17(6), 45-49.

- Heiskari, J., & Lehtola, L. (2009, 1-3 Dec. 2009). *Investigating the State of User Involvement in Practice*. Paper presented at the 16th Asia-Pacific Software Engineering Conference, Penang, Malaysia.
- Jackman, M. (1998). Homeopathic remedies for team toxicity. *IEEE Software*, 15(4), 43-45.
- Jiandong, Z. (2009, 11-13 Dec. 2009). *Employee Engagement Investigation in IT Industry*. Paper presented at the International Conference on Computational Intelligence and Software Engineering, Wuhan, China.
- Jones, C. (1995). How office space affects programming productivity. *Computer*, 28(1), 76-77.
- Jordan, G., & Segelod, E. (2006). Software innovativeness: outcomes on project performance, knowledge enhancement, and external linkages. *R&D Management*, 36(2), 127-142.
- Kieburtz, R. B., McKinney, L., Bell, J. M., Hook, J., Kotov, A., Lewis, J., et al. (1996, March 25-29). *A software engineering experiment in software component generation*. Paper presented at the 18th International Conference on Software Engineering, Berlin , Germany
- Lasserre, C. M., Acuña, S. T., & Quincoces, V. E. (2001, Nov. 6-8). *Empiric validation of the person to role allocation process*. Paper presented at the XXI Internatinal Conference of the Chilean Computer Science Society, Punta Arenas, Chile.
- Leonhard, C. A., & Davis, J. S. (1995). Job-shop development model: a case study. *IEEE Software*, 12(2), 86-92.
- Lim, W. C. (1994). Effects of reuse on quality, productivity, and economics. *IEEE Software*, 11(5), 23-30.
- Mackey, K. (1998). Beyond Dilbert: creating cultures that work [software industry]. *IEEE Software*, 15(1), 48-49.
- McConnell, S. (1998). Problem programmers. *IEEE Software*, 15(2), 126-128.

- Middleton, P., Ho Woo, L., & Irani, S. A. (2004). Why culling software colleagues is popular. *IEEE Software*, 21(5), 28-32.
- Mockus, A., Fielding, R. T., & Herbsleb, J. D. (2002). Two case studies of open source software development: Apache and Mozilla. *ACM Transactions on Software Engineering and Methodology*, 11(3), 309-346.
- Nelson, K. M., Nelson, H. J., & Ghods, M. (2001, Jan. 3-6). *Understanding the personal competencies of IS support experts: moving toward the e-business future*. Paper presented at the 34th Annual Hawaii International Conference on System Sciences, Maui, HI.
- Padberg, F., & Müller, M. M. (2003, Sept. 3-5). *Analyzing the cost and benefit of pair programming*. Paper presented at the Ninth International Software Metrics Symposium, Sydney, Australia.
- Patanakul, P. (2009, Aug. 2-6). *Toward an understanding of the dynamic of project manager assignments: An empirical study*. Paper presented at the International Conference on Management of Engineering & Technology, Portland, OR.
- Pillai, K., & Nair, V. S. S. (1998, March 26-28). *Early prediction of project schedule slippage*. Paper presented at the IEEE Workshop on Application-Specific Software Engineering Technology, Richardson, TX.
- Rauscher, T. G., & Smith, P. G. (1995). Time-Driven Development of Software in Manufactured Goods. *Journal of Product Innovation Management*, 12(3), 186-199.
- Rivera-Ibarra, J. G., Rodriguez-Jacobo, J., & Serrano-Vargas, M. A. (2010, March 9-12). *Competency Framework for Software Engineers*. Paper presented at the 23rd IEEE Conference on Software Engineering Education and Training, Pittsburgh, PA.
- Rothman, J. (1999). Retrain your code czar. *IEEE Software*, 16(2), 86-88.

- Sengupta, K., Abdel-Hamid, T. K., & Bosley, M. (1999). Coping with staffing delays in software project management: an experimental investigation. *IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans*, 29(1), 77-91.
- Shendil, K., & Madhavji, N. H. (1994, Oct. 5-7). *Personal 'progress functions' in the software process*. Paper presented at the Ninth International Software Process Workshop, Airlie, VA.
- Sneed, H. M. (1997, March 17-19). *Measuring the performance of a software maintenance department*. Paper presented at the First Euromicro Conference on Software Maintenance and Reengineering, Berlin , Germany
- Soedarsono, A. A., Murray, S. L., & Omurtag, Y. (1998). Productivity improvement at a high-tech state-owned industry-an Indonesian case study of employee motivation. *IEEE Transactions on Engineering Management*, 45(4), 388-395.
- Teasley, S., Covi, L., Krishnan, M. S., & Olson, J. S. (2000). *How does radical collocation help a team succeed?* Paper presented at the ACM conference on Computer supported cooperative work, Philadelphia, PA.
- Tomaszewski, P., & Lundberg, L. (2006). The increase of productivity over time--an industrial case study. *Information and Software Technology*, 48(9), 915-927.
- Williams, M., Packlick, J., Bellubbi, R., & Coburn, S. (2007, Aug. 13-17). *How We Made Onsite Customer Work - An Extreme Success Story*. Paper presented at the AGILE, Washington, DC.
- Zhong, X., Madhavji, N. H., & El Emam, K. (2000). Critical factors affecting personal software processes. *IEEE Software*, 17(6), 76-83.

Excluidos

- 199 *IEEE Power Engineering Society Summer Meeting. Conference Proceedings (Cat. No.99CH36364)*. (1999, 18-22 Jul 1999). Paper presented at the IEEE Power Engineering Society Summer Meeting.

Clinical Epilepsy: Pediatrics. (2005). *Epilepsia*, 46, 142-167.

Clinical Neurophysiology: EEG–Video Monitoring. (2005). *Epilepsia*, 46, 23-34.

Human Imaging: Adult. (2005). *Epilepsia*, 46, 35-48.

Non-AED/Non-Surgical Treatments (Hormonal, Ketogenic Diet, VNS, etc.): All Ages. (2005).
Epilepsia, 46, 226-235.

Nursing/Psychosocial/Health Services. (2005). *Epilepsia*, 46, 14-23.

Surgery: Pediatric. (2005). *Epilepsia*, 46, 247-253.

Monday, December 4, 2006 Poster Session III 7:30 a.m. – 4:30 p.m. (2006). *Epilepsia*, 47, 204-289.

Poster Session Monday July 3, 2006 13:30–15:00 Poster Session 1 Adult Epileptology. (2006).
Epilepsia, 47, 36-68.

Sunday, December 3, 2006 Poster Session II 7:30 a.m.–4:30 p.m. (2006). *Epilepsia*, 47, 119-204.

Wednesday July 5, 2006 13:30–15:00 Poster Session 3 Paediatric Epileptology. (2006).
Epilepsia, 47, 169-204.

ABSTRACTS. (2007). *Acta Pædiatrica*, 96, 46-61.

Abstracts of the 5th Joint Meeting of the German, Austrian, and Swiss Sections of the
International League Against Epilepsy Basle, May 16–19, 2007. (2007). *Epilepsia*, 48,
1-66.

Poster Session I 1:00 p.m.-6:00 p.m. (2007). *Epilepsia*, 48, 1-118.

Poster Session II 8:00 a.m.-6:00 p.m. (2007). *Epilepsia*, 48, 125-238.

Poster Session III 8:00 a.m.-2:00 p.m. (2007). *Epilepsia*, 48, 248-380.

Poster Sessions. (2007). *Epilepsia*, 48, 57-172.

Poster Session 2 11:00 a.m.-7:30 p.m. Professionals in Epilepsy care. (2008). *Epilepsia*, 49, 177-319.

Poster Session 3 8:00 a.m.-1:30 p.m. Translational Research. (2008). *Epilepsia*, 49, 337-475.

Saturday, December 6, 2008 Poster Session 1 1:00 p.m.-6:00 p.m. Clinical Neurophysiology. (2008). *Epilepsia*, 49, 1-166.

Abstracts from the Special Care Dentistry 21st Annual Meeting can be accessed in the electronic version of this issue at interscience.wiley.com. (2009). *Special Care in Dentistry*, 29(5), e1-e15.

Oral Communications. (2009). *Journal of Cardiovascular Electrophysiology*, 20, S1-S59.

Anwar, S., Ramzan, M., Rauf, A., & Shahid, A. A. (2010, April 21-23). *Software Maintenance Prediction Using Weighted Scenarios: An Architecture Perspective*. Paper presented at the International Conference on Information Science and Applications, Seoul, Korea.

Ballou, D. P., & Tayi, G. K. (1996). A decision aid for the selection and scheduling of software maintenance projects. *IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans*, 26(2), 203-212.

Barghouti, N. S. (1994, 5-7 Oct.). *Separating process model enactment from process execution in Provence*. Paper presented at the Ninth International Software Process Workshop Airlie, VA.

Bartram, D. (1987). The Development of an Automated Testing System for Pilot Selection: The MICROPAT Project1. *Applied Psychology*, 36(3-4), 279-298.

Basili, V. R., & Boehm, B. (2001). COTS-based systems top 10 list. *Computer*, 34(5), 91-95.

Blais, C. L. (1995, Dec. 3-6). *Scalability issues in enhancement of the MAGTF Tactical Warfare Simulation system*. Paper presented at the Simulation Conference, Arlington, VA.

- Borjesson, A., & Mathiassen, L. (2004). Successful process implementation. *IEEE Software*, 21(4), 36-44.
- Bowski, L., Perley, C. R., & West, J. M. (1983). A minicomputer system for analyzing and reporting pilot plant fermentor data. *Biotechnology and Bioengineering*, 25(5), 1237-1250.
- Butler, T. (2008). Teaching and Learning Guide for: Memoryscape: How Audio Walks Can Deepen Our Sense of Place by Integrating Art, Oral History and Cultural Geography. *Geography Compass*, 2(5), 1750-1754.
- Catal, C., & Diri, B. (2007, February 13-15). *Software defect prediction using artificial immune recognition system*. Paper presented at the IASTED International Conference on Software Engineering, Innsbruck, Austria.
- Chi, D. H., & Kuo, W. (1990). Optimal design for software reliability and development cost. *IEEE Journal on Selected Areas in Communications*, 8(2), 276-282.
- Chi, D. H., Lin, H. H., & Kuo, W. (1989, Jan. 24-26). *Software reliability and redundancy optimization*. Paper presented at the Annual Reliability and Maintainability Symposium, Atlanta, GA.
- Christian, J. T., & Eward, M. M. (1994, Oct. 11-13). *Transferring Software Engineering Technology: The Software Productivity Consortium Experience*. Paper presented at the IFIP TC8 Working Conference on Diffusion, Transfer, and Implementation of Information Technology, Pittsburgh, PA.
- Clark, B. K. (2000). Quantifying the effects of process improvement on effort. *IEEE Software*, 17(6), 65-70.
- Cunningham, T., Medlock, A., Sandefur, M., Rowse, J., & Asme. (2005, June 6-9). *High-productivity, low-cost, gas turbine system modeling software*. Paper presented at the ASME Turbo Expo, Reno-Tahoe, NV.

- Curtis, B., Hefley, W. E., Miller, S., & Konrad, M. (1997). Developing organizational competence. *Computer*, 30(3), 122-124.
- Cusick, J. J., & Ma, G. (2010, April 19-23). *Creating an ITIL inspired Incident Management approach: Roots, response, and results*. Paper presented at the IEEE/IFIP Network Operations and Management Symposium Workshops, Osaka, Japan.
- de Silva Garza, A. G., Franzoni Velázquez, A. L., & Morales, V. C. (2007). Promoting productivity in manufacturing companies in developing countries: An information system for managing and querying knowledge bases in the automotive industry in Mexico. *Information Technology for Development*, 13(3), 253-268.
- Di Penta, M., Casazza, G., Antoniol, G., & Merlo, E. (2001, March 14-16). *Modeling Web maintenance centers through queue models*. Paper presented at the Fifth European Conference on Software Maintenance and Reengineering, Lisbon , Portugal.
- Eilam, T., Kalantar, M. H., Konstantinou, A. V., Pacifici, G., Pershing, J., & Agrawal, A. (2006). Managing the configuration complexity of distributed applications in Internet data centers. *Communications Magazine, IEEE*, 44(3), 166-177.
- El Emam, K., & Birk, A. (2000). Validating the ISO/IEC 15504 measure of software requirements analysis process capability. *IEEE Transactions on Software Engineering*, 26(6), 541-566.
- Ericson, C. A. (2011). *System Safety Terms and Concepts*: John Wiley & Sons, Inc.
- Erlikh, L. (2000). Leveraging legacy system dollars for e-business. *IT Professional*, 2(3), 17-23.
- Fanberg, V. (2001, Oct. 8-10). *Use of binary file comparison tools in software release management*. Paper presented at the Second Asia-Pacific Conference on Quality Software, Hong Kong, China.
- Fazio, M. E. (2006, April 1-2). *Design and Development of a Model to Optimize the Clinical Engineering Department Size within a Hospital*. Paper presented at the IEEE 32nd Annual Northeast Bioengineering Conference, Easton, PA.

- Foley, D., & Calliss, F. W. (1993, Sept. 27-30). *Improving the quality of three products through improved testing: A case study*. Paper presented at the Conference on Software Maintenance, Montréal, Canada.
- Fullerton, N. (1998, May 21). *Energy Management System architecture*. Paper presented at the IEE Colloquium on The New NIE Energy Management System, Belfast , UK.
- Furth, R. (1996, April 21-24). *A model for professional training and education within a software engineering organization*. Paper presented at the Ninth Conference on Software Engineering Education, Daytona Beach, FL.
- Goh, T. N. (2010). An information management paradigm for statistical experiments. *Quality and Reliability Engineering International*, 26(5), 487-494.
- Gopal, A., & Koka, B. R. (2010). The Role of Contracts on Quality and Returns to Quality in Offshore Software Development Outsourcing. *Decision Sciences*, 41(3), 491-516.
- Guo, J., & Luqi. (2000, March 22-26). *A survey of software reuse repositories*. Paper presented at the Seventh IEEE International Conference and Workshop on the Engineering of Computer Based Systems, Edinburgh, UK.
- Harr, J., & Denault, G. (2002, April 15-19). *Issues concerning Linux clustering: cluster management and application porting*. Paper presented at the International Parallel and Distributed Processing Symposium, Fort Lauderdale, FL.
- Hartmann, J., & Robson, D. J. (1989, Oct. 16-19). *Revalidation during the software maintenance phase*. Paper presented at the Conference on Software Maintenance, Miami, FL.
- Hartmann, J., & Robson, D. J. (1990, Jan. 2-5). *RETEST-development of a selective revalidation prototype environment for use in software maintenance*. Paper presented at the Twenty-Third Annual Hawaii International Conference on System Sciences, Kailua-Kona, HI.

Heimbigner, D., & Wolf, A. L. (1996, June 17-19). *Software in the field needs process too*.

Paper presented at the 10th International Software Process Workshop, Dijon, France.

Holtzman De Garcia, L. N., Souza Tavares, S. R., & Segre, L. (1993, Dec. 17-18). *Participatory*

management: a case study in a high tech Brazilian plant. Paper presented at the IEEE

International Engineering Management Conference, New Delhi , India.

Irani, Z., & Sharp, J. M. (1997, 27-31 Jul 1997). *Re-engineering people through leadership and*

empowerment: a case study. Paper presented at the Portland International Conference

on Management and Technology Innovation in Technology Management, Portland,

OR.

Jai, A., & Sarkar, S. (2005, 03-06 Jan. 2005). *Staffing Software Maintenance and Support*

Projects. Paper presented at the 38th Annual Hawaii International Conference on

System Sciences, Big Island, HI.

Jiamthubthugsin, W., & Sutivong, D. (2006, 04-07 Jan. 2006). *Resource Decisions in Software*

Development Using Risk Assessment Model. Paper presented at the 39th Annual

Hawaii International Conference on System Sciences, Kauai, HI.

Jiang, X.-r., & Li, S.-c. (2010, 16-18 July 2010). *The antecedents of business-to business*

relational stability and its impacts on relational performance. Paper presented at the

IEEE International Conference on Software Engineering and Service Sciences, Beijing,

China.

Jones, C. (1994). Globalisation of software supply and demand. *Software Engineering Journal*,

9(6), 235-243.

Jones, C. (1995). End user programming. *Computer*, 28(9), 68-70.

Kazemzadeh, R. B., & Bashiri, M. (2005, Sept. 11-13, 2005). *Determination of critical factors*

on employee satisfaction - a case study on a financial institute. Paper presented at the

IEEE International Engineering Management Conference, St. John's, Newfoundland &

Labrador, Canada.

- Kazman, R., Klein, M., Barbacci, M., Longstaff, T., Lipson, H., & Carriere, J. (1998, Aug. 10-14). *The architecture tradeoff analysis method*. Paper presented at the IEEE International Conference on Engineering of Complex Computer Systems, Monterey, CA.
- Khoshgoftaar, T. M., Lanning, D. L., & Pandya, A. S. (1993, Nov. 3-6). *A neural network modeling methodology for the detection of high-risk programs*. Paper presented at the Fourth International Symposium on Software Reliability Engineering, Denver, CO.
- Killen, C. P. (1997, July 27-31). *The non-technical side of manufacturing change: Australian case studies*. Paper presented at the Portland International Conference on Management and Technology Innovation in Technology Management, Portland, OR.
- Kitsopoulos, S. C. (1994, Oct. 17-19). *From mentoring to executive coaching-a psychological case study*. Paper presented at the IEEE International Engineering Management Conference, Dayton North, OH.
- Knudstrup, J., Haggouchi, K., Peron, M., Quinn, P., Ballester, P., Banse, K., et al. (2002, Aug. 22). *Evolution and adaptation of the VLT data flow system*. Paper presented at the Observatory Operations to Optimize Scientific Return III, Waikoloa, HI.
- Krishnan, M. S., Kriebel, C. H., Kekre, S., & Mukhopadhyay, T. (2000). An empirical analysis of productivity and quality in software products. *Management Science*, 46(6), 745-759.
- Lacy, J. (1983). Preface. *IBM Systems Journal*, 22(3), 168-169.
- Laird, L. M. (2006). The Limitations of Estimation. *IT Professional*, 8(6), 40-45.
- Lilley, M. M., & Williams, R. G. (1991, Oct. 27-31). *Partner selection for joint venture agreements*. Paper presented at the Technology Management : the New International Language, Portland, OR.
- Lirov, Y., Prakash, S., & Ravikumar, S. (1991). ALCA — automated local area networks configuration aid. *Expert Systems*, 8(3), 171-182.

- Little, T. (2004). Value creation and capture: a model of the software development process. *IEEE Software*, 21(3), 48-53.
- Lynex, A., & Layzell, P. J. (1997, July 14-18). *Understanding resistance to software reuse*. Paper presented at the Eighth IEEE International Workshop on Software Technology and Engineering Practice, London , UK.
- Martin, S., & Kidwell, D. K. (2001, Aug. 16-17). *A case study in cluster analysis for intranet organization*. Paper presented at the 2nd International Workshop on Engineering Management for Applied Technology, Austin, TX.
- Mauceri, L. J. (1974). Control of an expanding network—"An operational nightmare". *Networks*, 4(4), 287-297.
- McGarry, F., & Decker, B. (2002). Attaining Level 5 in CMM process maturity. *IEEE Software*, 19(6), 87-96.
- Moore, B. J. (1994, Oct. 17-19). *Case study of unforeseen negative impacts of a TQM program*. Paper presented at the IEEE International Engineering Management Conference, Dayton North, OH.
- Ning, Z., Xiaoting, H., Xiaojun, Z., & Guangyu, X. (2009, Dec. 11-13). *Performance Appraisal System for Academic Staff in the Context of Digital Campus of Higer Education Institutions: Design and Implementation*. Paper presented at the International Conference on Computational Intelligence and Software Engineering, Wuhan, China.
- Ooi, G., & Soh, C. (2003). Developing an activity-based costing approach for system development and implementation. *SIGMIS Database*, 34(3), 54-71.
- Ortiz, E. A. (1995, Jan. 16-19). *Software-quality improvement using reliability-growth models*. Paper presented at the Annual Reliability and Maintainability Symposium, Washington, DC.
- Otero, C. E., Otero, L. D., Weissberger, I., & Qureshi, A. (2010, March 24-26). *A Multi-criteria Decision Making Approach for Resource Allocation in Software Engineering*.

Paper presented at the 12th International Conference on Computer Modelling and Simulation, Cambridge, United Kingdom.

Padberg, F. (2002). A discrete simulation model for assessing software project scheduling policies. *Software Process: Improvement and Practice*, 7(3-4), 127-139.

Padberg, F. (2002, Dec. 4-6). *Using process simulation to compare scheduling strategies for software projects*. Paper presented at the Ninth Asia-Pacific Software Engineering Conference, Queensland, Australia.

Parkan, C., Lam, K., & Hang, G. (1997). Operational competitiveness analysis on software development. *Journal of the Operational Research Society*, 48(9), 892-905.

Pei, G., & Victoria, A. (1994). Reusability in software maintenance: A Japan—USA comparison. *Journal of Software Maintenance: Research and Practice*, 6(4), 165-183.

Pfahl, D., Al-Emran, A., & Ruhe, G. (2006). Simulation-based stability analysis for software release plans. In Q. Wang, D. Pfahl, D. M. Raffo & P. Wernick (Eds.), *Software Process Change* (Vol. 3966, pp. 262-273): Springer Berlin / Heidelberg.

Pfahl, D., Al-Emran, A., & Ruhe, G. (2007). A System Dynamics Simulation Model for Analyzing the Stability of Software Release Plans. *Software Process: Improvement and Practice*, 12(5), 475-490.

Phister Jr, M. (1981). Computer Capacity Management. *Information & Management*, 4(5), 269-273.

Phister Jr, M. (1981). A model of the software development process. *Journal of Systems and Software*, 2(3), 237-255.

Prendergast, B. (1995, Nov. 13-15). *Inter-discipline cross-shift teams for station improvements-a case study*. Paper presented at the IEEE Advanced Semiconductor Manufacturing Conference and Workshop, Cambridge, MA.

- Qing, Z., & Zhiming, R. (2010, May 7-9). *Research on Human Resource Configuration Strategy in Software Engineering*. Paper presented at the International Conference on E-Business and E-Government, Guangzhou, China.
- Qingrui, X., Zhangshu, X., & Ling, Z. (2003, Nov. 2-4). *Innovation by everyone: case study from a Chinese top enterprise*. Paper presented at the IEEE International Engineering Management Conference, Estoril, Portugal.
- Ramesh, B., & Edwards, M. (1993, Jan. 4-6). *Issues in the development of a requirements traceability model*. Paper presented at the IEEE International Symposium on Requirements Engineering, San Diego, CA.
- Rauscher, T. G., & Smith, P. G. (1995). From experience time-driven development of software in manufactured goods. *Journal of Product Innovation Management*, 12(3), 186-199.
- Rivas, L., Perez, M., Mendoza, L. E., & Griman, A. (2008, Oct. 26-31). *Towards a Selection Model for Software Engineering Tools in Small and Medium Enterprises (SMEs)*. Paper presented at the Third International Conference on Software Engineering Advances, Sliema, Malta.
- Ropponen, J., & Lyytinen, K. (2000). Components of software development risk: how to address them? A project manager survey. *IEEE Transactions on Software Engineering*, 26(2), 98-112.
- Sadr, B., & Dousette, P. J. (1996). An OO project management strategy. *Computer*, 29(9), 33-38.
- Salmeron, J. L., & Lopez, C. (2010). A multicriteria approach for risks assessment in ERP maintenance. *Journal of Systems and Software*, 83(10), 1941-1953.
- Sanders, N. (1996, Jan. 24-27). *Automated testing using executable formal specifications*. Paper presented at the International Conference Software Engineering: Education and Practice, Dunedin, New Zealand.

- Sardos, P. A. (1994, Oct. 17-19). *Quality management-a case study*. Paper presented at the IEEE International Engineering Management Conference, Dayton North, OH.
- Selamat, M. H., Othman, A. T., Rahim, M. M., & Khalil, I. (1993, July 19-23). *A model for CASE implementation: A Malaysian experience*. Paper presented at the Sixth International Workshop on Computer-Aided Software Engineering, Kent Ridge, Singapore.
- Shepperd, M., Schofield, C., & Kitchenham, B. (1996, March 25-29). *Effort estimation using analogy*. Paper presented at the 18th International Conference on Software Engineering, Berlin, Germany.
- Sherer, S. W., Kouchakdjian, A., & Arnold, P. G. (1996). Experience using cleanroom software engineering. *IEEE Software*, 13(3), 69-76.
- Solotorevsky, G., Gudes, E., & Meisels, A. (1991, May 28-29). *Specifying resource allocation and time-tabling problems using a rule-based language*. Paper presented at the Fifth Israel Conference on Computer Systems and Software Engineering, Herzlia, Israel.
- Stark, G. E., & Oman, P. W. (1997). Software maintenance management strategies: observations from the field. *Journal of Software Maintenance: Research and Practice*, 9(6), 365-378.
- Strigel, W. (2002). *Software Productivity Center Inc*: John Wiley & Sons, Inc.
- Swedish, J. A. (1998, Dec. 13-16). *Simulation of an inland waterway barge fleet distribution network*. Paper presented at the Winter Simulation Conference, Washington, DC.
- Tai, A. T., Trivedi, K. S., & Hecht, H. (1997, Aug. 11-12). *On the development of dependability-evaluation workbench for high-assurance system designers*. Paper presented at the High-Assurance Systems Engineering Workshop, 1997., Proceedings.
- Taizan, C. (2000, Oct. 14). *Beyond productivity in software maintenance: factors affecting lead time in servicing users' requests*. Paper presented at the International Conference on Software Maintenance, San Jose, CA.

- Tamrat, E., Vilkinas, T., & Warren, J. R. (1996, Jan. 3-6). *Analysis of a telecommuting experience: a case study*. Paper presented at the 29th Hawaii International Conference on System Sciences, Maui, HI.
- Tan, T., Li, Q., Boehm, B., Yang, Y., He, M., & Moazeni, R. (2009). *Productivity trends in incremental and iterative software development*. Paper presented at the 3rd International Symposium on Empirical Software Engineering and Measurement, Lake Buena Vista, FL.
- Teng, Y., & He, Z. (2009, May 15-17). *Design and Implementation of Stored Procedure Router Based on Dynamic SQL*. Paper presented at the International Forum on Information Technology and Applications, Chengdu, China.
- Tsagias, M., & Kitchenham, B. (2000). An evaluation of the business object approach to software development. *Journal of Systems and Software*, 52(2-3), 149-156.
- Ureyen, R., Meydanli, I. I., & Koksaldi, S. (1999, July 25-29). *Reorganization of an industrial R&D center: a case study*. Paper presented at the Portland International Conference on Management of Engineering and Technology, Portland, OR.
- Vaishnavi, V. K., Abbott, D., Cooper, T. A., & Walling, K. (1992, Sep. 21-25). *Panel on management of emerging software technologies*. Paper presented at the Sixteenth Annual International Computer Software and Applications Conference, Chicago, IL.
- Velez, R., Zhang, D., & Kho, J. (1993, Nov. 8-11). *An intelligent tool for UNIX performance tuning*. Paper presented at the Fifth International Conference on Tools with Artificial Intelligence, Boston, MA.
- Vucetic, J., & Kline, P. (1998, May 18-20). *Network management applications for wireless local loop*. Paper presented at the 9th Mediterranean Electrotechnical Conference, Tel-Aviv, Israel.
- Wald, B., & Salisbury, L. A. (1977). The Computer Family Architecture Project: Service Perspectives and Overview. *Computer*, 10(10), 8-11.

- Wang, P., & Turban, E. (1994). Management information systems issues of the 1990s in the Republic of China: An industry analysis. *International Journal of Information Management*, 14(1), 25-38.
- Wiegner, R. T., & Nof, S. Y. (1993). The software product feedback flow model for development planning. *Information and Software Technology*, 35(8), 427-438.
- Yu, W. D., & Patil, G. (2007, July 1-4). *A Workflow-Based Test Automation Framework for Web Based Systems*. Paper presented at the 12th IEEE Symposium on Computers and Communications, Kerkyra, Greece.
- Zeng, Y., Wang, L., & Zhang, J. (2006, May 27-28). *Aggregative risk assessment model for information technology project development*. Paper presented at the Fifth Wuhan International Conference on E-Business, Wuhan, China.

Fase 1. Resultados tras lectura de texto completo

Excluidos

- Abdel-Hamid, T. K., Sengupta, K., & Hardebeck, M. J. (1994). The effect of reward structures on allocating shared staff resources among interdependent software projects: an experimental investigation. *IEEE Transactions on Engineering Management*, 41(2), 115-125.
- Abdel-Hamid, T. K., Sengupta, K., & Swett, C. (1999). The impact of goals on software project management: An experimental investigation. *MIS Quarterly*, 23(4), 531-555.
- Andersson, I., & Nilsson, K. (2002, January 7-10). *Improving diffusion practices in a software organization*. Paper presented at the 35th Annual Hawaii International Conference on System Sciences, Hawaii, HI.
- April, A. (2010, Sept. 29-Oct. 2). *Studying Supply and Demand of Software Maintenance and Evolution Services*. Paper presented at the Seventh International Conference on the Quality of Information and Communications Technology, Porto, Portugal.

- Barreto, A., Barros, M. d. O., & Werner, C. M. L. (2008). Staffing a software project: A constraint satisfaction and optimization-based approach. *Computers & Operations Research*, 35(10), 3073-3089.
- Ben-Bassat, M. (1999, 30 Aug.-2 Sept.). *Creating service expertise from raw data with expert system software*. Paper presented at the IEEE Systems Readiness Technology Conference, San Antonio, TX.
- Bertolino, A., Marchetti, E., & Mirandola, R. (2007). Performance measures for supporting project manager decisions. *Software Process: Improvement and Practice*, 12(2), 141-164.
- Bianchi, A., Lanubile, F., & Visaggio, G. (2001, April 4-6). *A controlled experiment to assess the effectiveness of inspection meetings*. Paper presented at the Seventh International Software Metrics Symposium, London, UK.
- Biffl, S., & Halling, M. (2003). Investigating the defect detection effectiveness and cost benefit of nominal inspection teams. *IEEE Transactions on Software Engineering*, 29(5), 385-397.
- Boehm, B. W., & Papaccio, P. N. (1988). Understanding and controlling software costs. *IEEE Transactions on Software Engineering*, 14(10), 1462-1477.
- Bommer, M., & Pease, V. (1991). Mitigating the impact of project cancellations on productivity. *National Productivity Review*, 10(4), 453-463.
- Canfora, G., Cimitile, A., Garcia, F., Piattini, M., & Visaggio, C. A. (2006, March 22-24). *Performances of pair designing on software evolution: a controlled experiment*. Paper presented at the 10th European Conference on Software Maintenance and Reengineering, Bari, Italy.
- Cao, L., Ramesh, B., & Abdel-Hamid, T. (2010). Modeling dynamics in agile software development. *ACM Transactions on Information Systems*, 1(1), 1-26.

- Chan, T. Z. (2000, Oct. 14). *Beyond productivity in software maintenance: Factors affecting lead time in servicing users' requests*. Paper presented at the International Conference on Software Maintenance, San Jose, CA.
- Chang, K.-c., Sheu, T. S., Klein, G., & Jiang, J. J. (2010). User commitment and collaboration: Motivational antecedents and project performance. *Information and Software Technology*, 52(6), 672-679.
- Diamant, E. I., Fussell, S. R., & Lo, F.-I. (2008). *Where did we turn wrong?: unpacking the effect of culture and technology on attributions of team performance*. Paper presented at the ACM conference on Computer supported cooperative work, San Diego, CA.
- Dong, F., Li, M., Li, J., Yang, Y., & Wang, Q. (2009, Oct. 15-16). *Effect of staffing pattern on software project: An empirical analysis*. Paper presented at the 3rd International Symposium on Empirical Software Engineering and Measurement, Lake Buena Vista, FL.
- Dura, O., & Yilmaz, A. E. (2009, Sept. 14-16). *Software product line development: A review on practical issues and challenges*. Paper presented at the 24th International Symposium on Computer and Information Sciences, North Cyprus, Turkey.
- Erdogmus, H. (2007, Sept. 20-21). *A cost effectiveness indicator for software development*. Paper presented at the First International Symposium on Empirical Software Engineering and Measurement, Madrid, Spain.
- Fægri, T. E., Dyba, T., & Dingsoyr, T. (2010). Introducing knowledge redundancy practice in software development: Experiences with job rotation in support work. *Information and Software Technology*, 52(10), 1118-1132.
- Frangos, S. A. (1998). Motivated humans for reliable software products. *Microprocessors and Microsystems*, 21(10), 605-610.
- Goebel, C. J. (2009, Aug. 24-28). *How Being Agile Changed Our Human Resources Policies*. Paper presented at the Agile Conference, Chicago, IL.

- Hale, J., Parrish, A., Dixon, B., & Smith, R. K. (2000). Enhancing the Cocomo estimation models. *IEEE Software*, 17(6), 45-49.
- Heiskari, J., & Lehtola, L. (2009, 1-3 Dec. 2009). *Investigating the State of User Involvement in Practice*. Paper presented at the 16th Asia-Pacific Software Engineering Conference, Penang, Malaysia.
- Jackman, M. (1998). Homeopathic remedies for team toxicity. *IEEE Software*, 15(4), 43-45.
- Jiandong, Z. (2009, 11-13 Dec. 2009). *Employee Engagement Investigation in IT Industry*. Paper presented at the International Conference on Computational Intelligence and Software Engineering, Wuhan, China.
- Jones, C. (1995). How office space affects programming productivity. *Computer*, 28(1), 76-77.
- Jordan, G., & Segelod, E. (2006). Software innovativeness: outcomes on project performance, knowledge enhancement, and external linkages. *R&D Management*, 36(2), 127-142.
- Lasserre, C. M., Acuña, S. T., & Quincoces, V. E. (2001, Nov. 6-8). *Empiric validation of the person to role allocation process*. Paper presented at the XXI Internatinal Conference of the Chilean Computer Science Society, Punta Arenas, Chile.
- Leonhard, C. A., & Davis, J. S. (1995). Job-shop development model: a case study. *IEEE Software*, 12(2), 86-92.
- Lim, W. C. (1994). Effects of reuse on quality, productivity, and economics. *IEEE Software*, 11(5), 23-30.
- Mackey, K. (1998). Beyond Dilbert: creating cultures that work [software industry]. *IEEE Software*, 15(1), 48-49.
- McConnell, S. (1998). Problem programmers. *IEEE Software*, 15(2), 126-128.
- Middleton, P., Ho Woo, L., & Irani, S. A. (2004). Why culling software colleagues is popular. *IEEE Software*, 21(5), 28-32.

- Mockus, A., Fielding, R. T., & Herbsleb, J. D. (2002). Two case studies of open source software development: Apache and Mozilla. *ACM Transactions on Software Engineering and Methodology*, 11(3), 309-346.
- Nelson, K. M., Nelson, H. J., & Ghods, M. (2001, Jan. 3-6). *Understanding the personal competencies of IS support experts: moving toward the e-business future*. Paper presented at the 34th Annual Hawaii International Conference on System Sciences, Maui, HI.
- Patanakul, P. (2009, Aug. 2-6). *Toward an understanding of the dynamic of project manager assignments: An empirical study*. Paper presented at the International Conference on Management of Engineering & Technology, Portland, OR.
- Pillai, K., & Nair, V. S. S. (1998, March 26-28). *Early prediction of project schedule slippage*. Paper presented at the IEEE Workshop on Application-Specific Software Engineering Technology, Richardson, TX.
- Rauscher, T. G., & Smith, P. G. (1995). Time-Driven Development of Software in Manufactured Goods. *Journal of Product Innovation Management*, 12(3), 186-199.
- Rivera-Ibarra, J. G., Rodriguez-Jacobo, J., & Serrano-Vargas, M. A. (2010, March 9-12). *Competency Framework for Software Engineers*. Paper presented at the 23rd IEEE Conference on Software Engineering Education and Training, Pittsburgh, PA.
- Rothman, J. (1999). Retrain your code czar. *IEEE Software*, 16(2), 86-88.
- Sengupta, K., Abdel-Hamid, T. K., & Bosley, M. (1999). Coping with staffing delays in software project management: an experimental investigation. *IEEE Transactions on Systems, Man and Cybernetics, Part A: Systems and Humans*, 29(1), 77-91.
- Shendil, K., & Madhavji, N. H. (1994, Oct. 5-7). *Personal 'progress functions' in the software process*. Paper presented at the Ninth International Software Process Workshop, Airlie, VA.

Sneed, H. M. (1997, March 17-19). *Measuring the performance of a software maintenance department*. Paper presented at the First Euromicro Conference on Software Maintenance and Reengineering, Berlin , Germany.

Soedarsono, A. A., Murray, S. L., & Omurtag, Y. (1998). Productivity improvement at a high-tech state-owned industry-an Indonesian case study of employee motivation. *IEEE Transactions on Engineering Management*, 45(4), 388-395.

Teasley, S., Covi, L., Krishnan, M. S., & Olson, J. S. (2000). *How does radical collocation help a team succeed?* Paper presented at the ACM conference on Computer supported cooperative work, Philadelphia, PA.

Tomaszewski, P., & Lundberg, L. (2006). The increase of productivity over time--an industrial case study. *Information and Software Technology*, 48(9), 915-927.

Williams, M., Packlick, J., Bellubbi, R., & Coburn, S. (2007, Aug. 13-17). *How We Made Onsite Customer Work - An Extreme Success Story*. Paper presented at the AGILE, Washington, DC.

Incluidos

Kieburtz, R. B., McKinney, L., Bell, J. M., Hook, J., Kotov, A., Lewis, J., et al. (1996, March 25-29). *A software engineering experiment in software component generation*. Paper presented at the 18th International Conference on Software Engineering, Berlin , Germany.

Padberg, F., & Müller, M. M. (2003, Sept. 3-5). *Analyzing the cost and benefit of pair programming*. Paper presented at the Ninth International Software Metrics Symposium, Sydney, Australia.

Zhong, X., Madhavji, N. H., & El Emam, K. (2000). Critical factors affecting personal software processes. *IEEE Software*, 17(6), 76-83.

Fase 2. Resultados tras lectura de texto completo

Excluidos

Basili, V. R., Briand, C. L., & Melo, W. L. (1996). How reuse influences productivity in object-oriented systems. *Communications of the ACM*, 39(10), 104-106.

Brooks, F. (1995). *The Mythical Man-Month: Essays on Software Engineering*. Reading, MA: Addison-Wesley.

DeMarco, T. (1982). *Controlling Software Projects: Management, Measurement & Estimation*. Englewood Cliffs, N.J.: Prentice-Hall.

Humphrey, W. S. (1995). Introducing the Personal Software Process. *Annals of Software Engineering*, 1, 311-325.

Sommerville, I. (1996). *Software Engineering*. Addison-Wesley.

Rout, T. P. (1995). SPICE: A Framework for Software Process Assessment. *Software Process: Improvement and Practice*, 1(1), 57-66.

Incluidos

Bisant, D., & Lyle, J. (1989). A two-person inspection method to improve programming productivity. *IEEE Transactions on Software Engineering*, 15(10), 1294-1304.

DeMarco, T., & Lister, T. (1985). *Programmer Performance and the Effects of the Workplace*. Paper presented at the 8th International Conference on Software Engineering. New York: Institute of Electrical and Electronics Engineers.

Sherdil, K., & Madhavji, N. H. (1996). *Human-Oriented Improvement in the Software Process*. Paper presented at the Fifth European Workshop on Software Process Technology.

Anexo C - Guión de entrevista semiestructurada

A continuación se incluye el guión base para las entrevistas realizadas en la fase cualitativa del enfoque del problema. Hay que tener en cuenta que la entrevista es de tipo semiestructurada con lo cual no es un guión cerrado ni limitado a este contenido.

Presentación

- Buenos días/tardes, muchas gracias por venir. ¿Le ha resultado difícil encontrarme? ¿Qué tal el día? (*otras preguntas para “romper el hielo”*²⁸).
- Es importante que sepa que el contenido de la entrevista será grabado, tal y como le informamos anteriormente (*enseñar hoja firmada por la persona*). De modo que posteriormente, la conversación será transcrita y cualquier nombre personal incluido en la misma será sustituido por iniciales no identificativas. En su caso, por ejemplo (*nombre*) será cambiado por (*inicial distinta del nombre*).
- *Sólo para programadores, analistas y jefes de proyecto:*
 - Cuéntenos de manera resumida en qué consiste su trabajo diario en el proyecto.

²⁸ El contenido en cursiva son aclaraciones para el entrevistador, el resto son frases literales que el entrevistador puede cambiar si considera oportuno pero siempre respetando la finalidad de cada frase.

Temas generales

- ¿Puedes darme una definición de la productividad a nivel de la organización?
¿Y a nivel de proyecto?
 - *Sólo para programadores, analistas y jefes de proyecto:*
 - ¿Y a nivel individual?
 - *Sólo para comerciales y clientes:*
 - ¿Percibe de alguna manera las diferencias en la productividad de los proyectos de desarrollo software? ¿Cómo? ¿De qué manera?
- En su opinión, ¿qué elementos indican productividad? ¿Por qué?
- *Sólo para programadores, analistas y jefes de proyecto:*
 - ¿Usan algún sistema para computar las horas de trabajo realizadas?
¿Las asignan de alguna manera al proyecto o tareas, o son asignadas de manera global? ¿Conoce la utilización de esta información?

Temas específicos

- Desde su punto de vista, ¿cuáles cree que son los factores que influyen en el proceso productivo?
 - *Sólo para programadores, analistas y jefes de proyecto:*
 - ¿Y en su trabajo diario?
- ¿Cómo cree que influyen cada uno de estos factores en la productividad? ¿De manera indirecta o directa? (*Utilizar esta pregunta sólo si el entrevistado no da información relevante sobre la influencia de los factores*).
 - *Sólo para programadores, analistas y jefes de proyecto:*
 - ¿Qué elementos produce en su trabajo? ¿Considera que existe alguna dificultad para medir estos elementos?

- ¿Qué elementos son elementos necesarios en su trabajo?
¿Qué necesita para realizar sus tareas? ¿Considera que existe alguna dificultad para medir estos elementos?
- ¿Conoce las medidas basadas en líneas de códigos y puntos de función?
 - *Si la respuesta es afirmativa:*
 - ¿En qué medida cree que sirven para medir su trabajo diario?
- *Sólo para comerciales y clientes:*
 - ¿Qué elementos considera que son producidos en los proyectos de desarrollo software? ¿Considera que existe alguna dificultad para medir alguno de estos elementos? (*Si la respuesta es afirmativa, ¿cuáles y qué dificultades considera que existen?*)
 - ¿Qué elementos considera que son necesarios en la realización de un desarrollo software? ¿Considera que existe alguna dificultad para medir estos elementos?
- *Sólo para programadores, analistas y jefes de proyecto:*
 - ¿Mide alguien la productividad de su trabajo?
 - *Si la respuesta es reflexiva, es decir, el entrevistado tiene una medida de su productividad, entonces:*
 - ¿Qué dificultades se encuentra al medir la productividad? ¿Surge algún enfrentamiento?
 - *En caso contrario:*
 - ¿Por qué motivos considera que no se está midiendo la productividad? ¿Ha habido algún problema anteriormente?

Satisfacción laboral

- ¿Está usted satisfecho con el tipo de trabajo que realiza? ¿Por qué?
- ¿Está usted satisfecho con los empleados que dependen de usted? ¿Y con sus jefes y superiores? ¿Por qué?
- ¿Y con sus compañeros? ¿Por qué?
- ¿Está usted satisfecho con la organización del trabajo que actualmente tiene? ¿Por qué?
- ¿Qué haría mejorar tu satisfacción laboral?
- ¿Qué haría mejorar tu productividad?

Cierre

- ¿Le gustaría contarnos algo que no hallamos tratado y que considere relevante para el tema que estamos tratando de estudiar? ¿De qué se trata?
- Muchas gracias acceder a realizar esta entrevista y por participar en el proyecto. La información que extraigamos del contenido de la entrevista nos permitirá continuar con el proyecto.
- Antes de finalizar la entrevista, nos gustaría comentarle que, tras la realización del proyecto, podrá acceder a los resultados del mismo, así como a las publicaciones que se produzcan a partir de los datos recogidos en la misma. Seguramente serán de gran interés para usted.

Anexo D - Cuestionario y justificación de selección de elementos

Cuestionario

A continuación se incluye una reproducción del cuestionario empleado en la fase cualitativa del enfoque del problema. En primer lugar se introduce el motivo del cuestionario explicando brevemente el objetivo del mismo además de mencionar la duración estimada del mismo. En segundo lugar, se pregunta al participante por tres datos demográficos: sexo, edad y modo de acceso al formulario. En tercer lugar, se pregunta al participante si tiene experiencia en cada uno de los puestos seleccionados, por este orden: programador, analista, consultor, y jefe de proyecto. En caso de tener experiencia, se requiere al participante valorar el grado de utilización de 16 elementos (*entradas*) y el grado de producción de otros 16 elementos (*salidas*). El cuestionario es accesible (actualmente) en la siguiente dirección web: <http://goo.gl/u62uq>

¿Qué utilizas y qué produces en tu trabajo?

El objetivo de esta encuesta es conocer qué utilizas y qué produces en tu trabajo. La encuesta es una necesidad dentro de la tesis doctoral, de D. Adrián Hernández López que está realizando en la UC3M bajo la dirección de D. Ricardo Colomo Palacios, que tiene como meta establecer nuevas medidas de productividad a nivel de trabajo. De forma general se puede decir que para desempeñar las funciones de cualquier puesto de trabajo son necesarios unos recursos; y a su vez, en cualquier puesto de trabajo se producen una serie de productos y/o servicios. Así pues, para

establecer nuevas medidas de productividad específicas de cada puesto es necesario conocer los recursos que utilizan y qué producen los trabajadores. Esto hace que necesitemos su participación para conocer qué necesita y qué produce en su puesto de trabajo. A continuación se presentan varias preguntas en función de diversos puestos de trabajo en los cuales usted desempeña sus funciones y/o tiene experiencia en ellos. La dedicación estimada para rellenar la encuesta varía entre 5 y 12 minutos.

Información demográfica

A. *Sexo*

- Hombre
- Mujer

B. *Edad*

¿Cómo ha accedido a esta encuesta? Seleccione el medio por el que ha accedido a esta encuesta

- He recibido un email directamente de los investigadores
- He recibido un email de una persona conocida
- He accedido desde una página web
- He accedido desde una red social

Si ha accedido mediante una página web, indique su dirección web.....

Si ha accedido mediante una red social, indique su nombre.....

Puestos

Programador

La función del programador es construir el código que dará lugar al producto resultante en base al diseño técnico, generando también el código asociado a los

procedimientos de migración y carga inicial de datos. Igualmente se encarga de la realización de las pruebas unitarias y participa en las pruebas de conjunto de la aplicación.

¿Realiza tareas actualmente como Programador y/o tiene experiencia en ellas?

- Sí
- No

Analista

La responsabilidad de los Analistas es elaborar un catálogo detallado de requisitos que permita describir con precisión el sistema de información, para lo cual mantendrán entrevistas y sesiones de trabajo con los responsables de la organización y usuarios, actuando del interlocutor entre estos y el equipo de proyecto en lo que a requerimientos se refiere. Estos requisitos permiten a los analistas elaborar los distintos modelos que sirven de base para el diseño, obteniendo los modelos de datos y de procesos en el caso del análisis estructurado y los modelos de clases e interacción de objetos en análisis orientado a objeto. Así mismo realizan la especificación de las interfaces entre el sistema y el usuario.

¿Realiza tareas actualmente como analista o tiene experiencia en ellas?

- Sí
- No

Consultor

La principal función de los Consultores es asesorar en las cuestiones sobre las que tienen un conocimiento especializado. Se diferencia así entre Consultor, que asesora en los aspectos relativos al negocio y Consultor Informático, con un nivel de especialización mayor en los aspectos relacionados con la informática, su aplicación e

integración en la organización. En el ámbito de la Consultoría Informática se distingue entre Tecnologías de la Información y Sistemas de Información. El Consultor en Tecnologías de la Información aporta un mayor conocimiento de las últimas tecnologías, colabora en la evaluación de distintas alternativas tecnológicas y participa en la validación y selección de la solución más adecuada para el sistema a desarrollar, mientras que el Consultor de Sistemas de Información ofrece una opinión experta, pericia o conocimientos relativos a los requisitos del negocio, técnicos y de usuario que han de tenerse en cuenta en el desarrollo de un sistema de información.

¿Realiza tareas actualmente como consultor y/o tiene experiencia en ellas?

- Sí
- No

Jefe de Proyecto

El Jefe de Proyecto realiza la estimación del esfuerzo necesario para llevar a cabo el proyecto, selecciona la estrategia de desarrollo, determina la estructura del mismo, fija el calendario de hitos y entregas y establece la planificación del proyecto. Es el encargado de dirigir el proyecto, realizando las labores de seguimiento y control del mismo, revisión y evaluación de resultados y coordinación del equipo de proyecto. Se ocupa también de la gestión y resolución de incidencias que puedan surgir durante el desarrollo del proyecto así como de la actualización de la planificación inicial. Entre sus funciones se encuentran la elaboración de los informes de seguimiento y el archivo de la documentación de gestión del proyecto una vez que este ha finalizado.

¿Realiza tareas actualmente como Jefe de Proyecto o tiene experiencia en ellas?

- Sí
- No

Entradas

¿Con qué frecuencia utiliza estos recursos para el puesto?

Marque con que frecuencia utiliza los siguientes recursos para desempeñar las funciones de Programador.

Nota: Se presentan los 16 elementos de entrada con una escala Likert de 6 valores: ("Nunca", "Casi Nunca", "A veces", "A menudo", "Casi Siempre", "Siempre". Los elementos son: *Tiempo, Conocimiento, Planificación, Estimación, Objetivos asignados, Software, Hardware, Instalaciones de trabajo, Especificación*

Salidas

¿Con que frecuencia produce estos productos/servicios para el puesto?

Marque con que frecuencia produce los siguientes productos/servicios para desempeñar las funciones de Programador.

Nota: Se presentan los 16 elementos de salida con una escala Likert de 6 valores: ("Nunca", "Casi Nunca", "A veces", "A menudo", "Casi Siempre", "Siempre". Los elementos son: *Producto, Documentación, Tarea finalizada, Cumplimiento de objetivos, Estimación, Planificación, Calidad, Ventas, Pruebas, Experiencia,*

Justificación de selección de elementos

La selección de la metodología y de las herramientas de recogida de información parte de las hipótesis establecidas. En este caso, la metodología tiene el objetivo de permitir el contraste de cuatro hipótesis que pueden agruparse en dos grupos: un primer grupo con las hipótesis de existencia de otras entradas (H1) y otras salidas (H3), de las habitualmente utilizadas en las medidas de productividad, y un segundo grupo con las hipótesis de diferenciación de la utilización de estas entradas (H2) y la

producción de estas salidas (H4) en función del puesto de trabajo. Así pues el propósito es doble: (1) comprobar la existencia de elementos, y (2) comprobar que hay diferencias entre el uso/producción de estos elementos en función del puesto de trabajo. Teniendo este doble propósito en mente se establecieron las pruebas estadísticas necesarias para contrastar las hipótesis tal y como se describe en el apartado 3.4.1 Metodología.

Dentro de la metodología de investigación planteada, se encuentra la construcción del formulario en la que se realizaron una serie de decisiones, siempre en base a conseguir el propósito investigador: el contraste de las hipótesis. Para ello se realizó la selección de los elementos a evaluar como entradas y salidas, y la escala de los mismos. Por un lado, se seleccionó la escala de tipo Likert, dada su fácil utilización, tanto para el investigador como para los participantes, y su rápida construcción. Por otro lado, la selección de los elementos parte de los datos obtenidos en la fase cualitativa previamente realizada, tal y como se explica a continuación.

La selección de los elementos se realizó de la siguiente manera:

1. Se incluyó el elemento "utilizado" en las medidas de productividad dentro de la categoría. Esto es, el tiempo en las entradas, y la funcionalidad y código fuente en las salidas.
2. Se incluyeron los elementos más mencionados en la fase cualitativa dentro de cada categoría. La media de menciones para las entradas fue de 3,442 ($\sigma = 2,879$) y para las salidas de 2,848 ($\sigma = 2,108$), por lo que los elementos por encima de esas medias son los considerados más mencionados. De este modo se incluyeron (de cada elemento se indican el número de participantes que las mencionó en la fase cualitativa):

Entradas:

- *Tiempo* - 14.

- *Cliente* - 11.
- *Especificación de requisitos* - 10.
- *Planificación* - 9.
- *Conocimiento* - 8.
- *Documentación* - 8.
- *Experiencia* - 8.
- *Software* - 8.
- *Hardware* - 7.
- *Objetivos asignados* - 6.
- *Estimación* - 4.

Salidas:

- *Documentación* - 13.
- *Código fuente* - 10.
- *Producto* - 9.
- *Conocimiento* - 9.
- *Calidad* - 8.
- *Ventas* - 7.
- *Tarea finalizada* - 7.
- *Solución de problemas* - 6.
- *Solución de bugs* - 6.
- *Pruebas* - 5.
- *Estimación* - 4.
- *Planificación* - 3.
- *Satisfacción del cliente* - 3.
- *Funcionalidad* - 3.

En este punto se tienen 11 entradas y 14 salidas, por lo que se decide fijar un número igual de entradas y de salidas, sin ninguna justificación específica, simplemente para que el número sea el mismo.

3. Finalmente se incluyeron una serie de elementos, que pese a que no se encuentran entre los elementos más mencionados, tienen unas peculiaridades por las que se les seleccionó. A continuación se incluyen estos elementos, indicando las menciones en la fase cualitativa, y la razón por la que se incluye cada uno de ellos:

Entradas:

- *Formación* - 3. Este elemento está ligeramente por debajo de la media pero se incluye ya que es un requisito omnipresente en las definiciones de puestos de trabajo (Fernández-Ríos, 1995). Además, la formación es un factor que afecta a la productividad de los trabajadores en IS (Wagner & Ruhe, 2008a).
- *Código fuente previo* - 2. Se incluye este elemento para que la propiedad de reutilización de software, así como las tareas de mantenimiento, estén presentes en las entradas consideradas. Por un lado, la reutilización es considerada como un elemento clave para aumentar la productividad (Gaffney, 1989; Glass, 1998); y por otro lado, las tareas de mantenimiento representan una fuerte carga de trabajo dentro de los proyectos de desarrollo de software (Banker, et al., 1991).
- *Instalaciones de trabajo* - 2. Este elemento se incluye dado que las características del lugar de trabajo pueden influir en la productividad de los trabajadores de IS (Burn, et al., 1991; Frangos, 1997; Gambill, et al., 2000; Garza, et al., 2003; Jones, 1995; Richens, 1998; H. I. Rubin & Hernandez, 1988; Sommerville, 2010)
- *Conocimiento funcional* - 2. Esta entrada se incluye como elemento especializado del conocimiento (Colombo, Mosca,

& Sartori, 2007) que ya ha sido incluido como entrada, y considerando que este elemento es crítico dentro del tipo de puestos de trabajo considerados en esta investigación (GUFPI-ISMA, 2012; Lee, Trauth, & Farwell, 1995).

- *Motivación*. Pese a no ser mencionada por los participantes se incluye esta entrada debido a la importancia que se le atribuye con respecto a la productividad (Beecham, et al., 2008; Paiva, et al., 2010; Trendowicz & Münch, 2009).

Salidas:

- *Cumplimiento de objetivos* - 2. Este elemento se incluye ya que entre las entradas más mencionadas, y por lo tanto entre las seleccionadas, se encuentran los *objetivos asignados*. Así pues, esta salida representa el resultado obtenido en función de dicha asignación.
- *Experiencia* - 1. Esta salida se incluye para que la formación de tipo *learn by doing* esté presente; esta formación es distinta al aprendizaje obtenido mediante una formación *formal*. Además, dado que se incluyó como entrada la *formación*, la inclusión de la experiencia representa una salida vinculada tanto con la producción, como con dicha formación recibida. A su vez, esta salida se menciona como factor incidente en la productividad en diversos estudios (ver Tabla 3).

Así pues, se incluyen finalmente 16 entradas y 16 salidas. Hay que tener en cuenta que estas entradas y salidas no son las únicas posibles, ni han de ser las más utilizadas y producidas. Con respecto a que no son las únicas posibles, basta con tener en cuenta que la listas de entradas y salidas obtenidas en la fase cualitativa incluye más elementos (52 entradas y 46 salidas) por lo que las seleccionadas son un conjunto reducido de esta que incluye algún elemento no mencionado. Con respecto a que no

han de ser las más utilizadas y producidas, es importante destacar, aunque parezca repetitivo, el objetivo de este cuestionario: contrastar las hipótesis planteadas. De este modo, y teniendo en cuenta que las hipótesis H1 y H3 plantean la existencia de otras entradas (H1) y otras salidas (H3) de las habitualmente consideradas en las medidas de productividad, la selección de un conjunto de entradas y salidas es válida siempre que incluye los elementos habitualmente considerados junto con otros elementos potencialmente utilizables. Este es el caso de los elementos seleccionados, tal y como se ha justificado anteriormente. Con respecto a las otras dos hipótesis a contrastar, se plantea que el uso de las entradas (H2) y la producción de las salidas (H4) difiere en función del puesto de trabajo, para ello se incluyen entradas que podrían ser utilizadas y salidas que podrían ser producidas en todos los puestos de trabajo, junto con entradas y salidas más específicas de determinados puestos (p. ej. *código fuente* o *estimación* como salidas, y *cliente* o *código fuente previo* como salidas).

Además, es importante destacar que el número de elementos (16) es pseudoaleatorio ya que como se mencionó anteriormente, se fijó que el número de entradas y de salidas fuera el mismo; se partió de las 14 salidas más utilizadas (incluyendo *código fuente* y *funcionalidad*) y finalmente se fijaron 16 elementos. Este número podría haber sido mayor o menor pero no influye en la eficacia del método planteado, ya que permite realizar los contrastes de las hipótesis planteadas, es decir, permite conseguir el objetivo por el que se plantea este método. No obstante, hay que destacar que un número menor de elementos hubiera reducido el tiempo de rellenado del cuestionario; lo que hubieran agradecido ciertos participantes que indicaron que la longitud del mismo era elevada, sobretodo en casos de participantes que rellenaron más de dos puestos de trabajo.

Anexo E - Condiciones para aplicar ANOVA

Para la utilización de pruebas paramétricas, como es el caso de ANOVA, es necesario que los datos cumplan los siguientes requisitos (Erceg-Hurn & Mirosevich, 2008):

- **Variable numérica.** En el caso de las variables incluidas en la fase cuantitativa para medir las entradas y salidas, la escala empleada es de tipo Likert. En determinadas circunstancias puede considerarse esta escala como de tipo intervalo, siempre y cuando se cumpla como requisito la igualdad de distancia entre valores de la escala. En este caso se va a suponer que la distancia entre valores de la escala es la misma y se da paso a comprobar los siguientes requisitos.
- **Normalidad.** Los valores de la variable dependiente han de seguir una distribución normal, al menos en la población en la que pertenece la muestra. Para comprobar este requisito se aplica la prueba estadística de Kolmogorov-Smirnov (Massey, 1951). Esta prueba permite comparar la distribución de la muestra con respecto a las distribuciones normal, exponencial, uniforme y poisson. En este caso se ha realizado el contraste con la distribución normal.
- **Homocedasticidad.** Las pruebas de homogeneidad de varianzas comprueban si las varianzas en las muestras son parecidas, es decir, si se cumple la condición de homocedasticidad. Existen varias pruebas para realizar la comprobación (p. ej., F de Fisher, Fmax de Hartley o prueba de Barlett). En esta ocasión se ha seleccionado la prueba de Levene (1960).

Estos requisitos suelen no ser comprobados en numerosas investigaciones (Keselman et al., 1998), lo que se traduce en errores de investigación (Erceg-Hurn & Mirosevich, 2008). En esta investigación, tras comprobar que ningún elemento cumplen los requisitos de normalidad y homocedasticidad se descarta el uso de pruebas paramétricas como estadísticos de contraste, y se opta por emplear un método no paramétrico, en concreto Kruskal Wallis (Kruskal & Wallis, 1952). No obstante, algunos elementos cumplen el requisito de homocedasticidad, aunque no es suficiente para aplicar métodos paramétricos. A continuación se presentan una tabla que indica si los elementos cumplen o no los requisitos, y posteriormente se muestran los resultados de la aplicación de cada una de las pruebas.

	Elemento	Normalidad	Homocedasticidad	¿Podrían aplicarse pruebas paramétricas?
Entradas	Tiempo	NO	SI	NO
	Conocimiento	NO	SI	NO
	Planificación	NO	NO	NO
	Estimación	NO	NO	NO
	Objetivos asignados	NO	NO	NO
	Software	NO	NO	NO
	Hardware	NO	SI	NO
	Instalaciones de trabajo	NO	SI	NO
	Especificación de requisitos	NO	SI	NO
	Conocimiento funcional	NO	SI	NO
	Cliente	NO	SI	NO
	Motivación	NO	SI	NO
	Documentación	NO	SI	NO
	Experiencia	NO	SI	NO
	Formación	NO	SI	NO

	Elemento	Normalidad	Homocedasticidad	¿Podrían aplicarse pruebas paramétricas?
	Código fuente previo	NO	NO	NO
Salidas	Código fuente	NO	NO	NO
	Producto	NO	NO	NO
	Documentación	NO	SI	NO
	Tarea finalizada	NO	NO	NO
	Cumplimiento de objetivos	NO	SI	NO
	Estimación	NO	SI	NO
	Planificación	NO	SI	NO
	Calidad	NO	SI	NO
	Ventas	NO	SI	NO
	Pruebas	NO	NO	NO
	Experiencia	NO	SI	NO
	Conocimiento	NO	SI	NO
	Solución de problemas	NO	NO	NO
	Solución de bugs	NO	NO	NO
	Satisfacción del cliente	NO	NO	NO
	Funcionalidad	NO	NO	NO

Tabla 52. Resultados de pruebas pre-requisito para aplicar pruebas paramétricas

Normalidad

Para comprobar el requisito de normalidad se emplea la prueba de Kolmogorov-Smirnov con contraste de distribución normal (Massey, 1951), y se han obtenido los siguientes resultados:

	Elemento	Kolmogorov-Smirnov Z	Asymp. Sig. (2-tailed)
Entradas	Tiempo	5,566	,000
	Conocimiento	6,007	,000
	Planificación	4,271	,000
	Estimación	3,881	,000
	Objetivos asignados	4,621	,000
	Software	4,107	,000
	Hardware	2,955	,000
	Instalaciones de trabajo	3,695	,000
	Especificación de requisitos	4,345	,000
	Conocimiento funcional	4,541	,000
	Cliente	3,774	,000
	Motivación	3,838	,000
	Documentación	4,388	,000
	Experiencia	4,379	,000
	Formación	3,913	,000
	Código fuente previo	3,008	,000
Salidas	Código fuente	3,425	,000
	Producto	3,521	,000
	Documentación	4,588	,000
	Tarea finalizada	4,760	,000
	Cumplimiento de objetivos	4,828	,000
	Estimación	3,896	,000
	Planificación	4,435	,000
	Calidad	4,436	,000
	Ventas	3,176	,000
	Pruebas	3,471	,000
	Experiencia	4,450	,000
	Conocimiento	4,369	,000
	Solución de problemas	4,168	,000
	Solución de bugs	2,527	,000
	Satisfacción del cliente	4,690	,000

	Elemento	Kolmogorov-Smirnov Z	Asymp. Sig. (2-tailed)
	Funcionalidad	4,650	,000

Tabla 53. Prueba de Kolmogorov-Smirnov (Normalidad)

Todos los elementos tienen un nivel de significancia menor de ,05, por lo que se cumple la hipótesis alterna que indica que la distribución de los datos difiere de la distribución normal; es decir, los elementos analizados no tienen distribución normal, y por ello no es posible aplicar sobre ellos pruebas paramétricas, entre ellas ANOVA.

Homocedasticidad

Para comprobar el requisito de homocedasticidad se ha empleado la prueba de Levene (1960), y se han obtenido los siguientes resultados:

	Elemento	Levene	Sig.
Entradas	Tiempo	,970	,407
	Conocimiento	,570	,635
	<i>Planificación</i>	3,452	,017*
	<i>Estimación</i>	2,976	,032*
	<i>Objetivos asignados</i>	3,681	,012*
	<i>Software</i>	9,885	,000*
	Hardware	,246	,864
	Instalaciones de trabajo	,554	,646
	Especificación de requisitos	2,591	,053
	Conocimiento funcional	,824	,481
	Cliente	1,839	,140
	Motivación	1,137	,334
	Documentación	,335	,800
	Experiencia	,784	,503
	Formación	,406	,749
	<i>Código fuente previo</i>	4,201	,006*
Salidas	<i>Código fuente</i>	12,956	,000*

Elemento	Levene	Sig.
<i>Producto</i>	6,449	,000*
Documentación	1,221	,302
<i>Tarea finalizada</i>	3,678	,012*
Cumplimiento de objetivos	1,565	,198
Estimación	1,787	,149
Planificación	1,329	,265
Calidad	2,448	,064
Ventas	2,198	,088
<i>Pruebas</i>	3,748	,011*
Experiencia	,109	,955
Conocimiento	1,929	,125
<i>Solución de problemas</i>	7,135	,000*
<i>Solución de bugs</i>	3,450	,017*
<i>Satisfacción del cliente</i>	4,255	,006*
<i>Funcionalidad</i>	4,201	,006*

Tabla 54. Prueba de Levene (Homocedasticidad)

Los elementos marcados con * presentan un nivel de significancia menor de ,05 por lo que su valor p es significativo, es decir, para estos elementos no se puede asumir la similitud de varianzas. En concreto hay 13 elementos (5 entradas y 8 salidas) que no cumplen el requisito de homocedasticidad. Así pues, no se cumple el requisito de homocedasticidad para todos los elementos y no es posible utilizar ANOVA sobre la totalidad de la muestra.

Anexo F - Publicaciones relacionadas con la tesis

2013

Hernández-López, A., Colomo-Palacios, R., & García-Crespo, A. (2013). Software Engineering Job Productivity: A systematic Review. *International Journal of Software Engineering and Knowledge Engineering*, 23(3), 387-406. doi: 10.1142/S0218194013500125 (JCR Impact factor 2012: 0.295; COMPUTER SCIENCE, SOFTWARE ENGINEERING, 96/105, Q4)

2012

Hernández-López, A. (2012). Satisfaction and Motivation: IT Practitioners' Perspective. *International Journal of Human Capital and Information Technology Professionals (IJHCITP)*, 3(4), 39-56. doi: 10.4018/jhcitp.2012100104 (Scopus Source Normalized Impact per Paper, SNIP 2012, 0.411)

Hernández-López, A., Colomo-Palacios, R., & García-Crespo, Á. (23/06/2012). Productivity in software engineering: a study of its meanings for practitioners. Artículo presentado en la 7th Iberian Conference on Information Systems and Technologies (CISTI), Madrid, Spain.

Hernández-López, A., & Colomo-Palacios, R. (13/06/2012). Job Satisfaction and Motivation of Software Engineering Practitioners. Artículo presentado en el

First Workshop on Managing the Influence of People and Team Factors in Software Engineering (INTEAMSE 2012), Madrid, Spain.

2011

Hernández-López, A., Colomo-Palacios, R., & García-Crespo, Á. (2011). Medición de la Productividad de los Puestos de Trabajo en Ingeniería del Software. *Revista Procesos y Métricas*, 8(1), 44-58.

Casado-Lumbreras, C., Colomo-Palacios, R., Hernández-López, A., & Soto-Acosta, P. (2011). Personnel Performance Appraisal Coverage in ITIL, COBIT and CMMi: A Study from the Perspective of People-CMM. *International Journal of Knowledge Society Research*, 2(2), 61-72. doi: 10.4018/jksr.2011040106

Colomo-Palacios, R., Hernández-López, A., García-Crespo, Á., & Paniagua-Martín, F. (2009, 7-9 October). *Personnel Performance Appraisal in ICT. A review of governance and maturity models*. Artículo presentado en la Conference on ENTERprise Information Systems, Ofir, Portugal.

Hernández-López, A., Colomo-Palacios, R., García-Crespo, Á., & Cabezas-Isla, F. (2011). Present, Past and Future of IT Careers, a Review: From the Local Pyramid to the "Flat World". In J. Luftman (Ed.), *Managing IT Human Resources: Considerations for Organizations and Personnel* (pp. 218-243). Hershey, PA: IGI Global. doi: 10.4018/978-1-60960-535-3.ch014

Colomo-Palacios, R., & Hernández-López, A. (2011). Improving IT Project Outcomes by Considering Alternative Perspectives. *International Journal of Information Technology Project Management*, 2(1), I-IV.

Hernández-López, A., Colomo-Palacios, R., García-Crespo, Á., & Cabezas-Isla, F. (2011). Software Engineering Productivity: Concepts, Issues and Challenges. *International Journal of Information Technology Project Management*, 2(1), 37-47. doi: 10.4018/jitpm.2011010103

2010

- Colomo-Palacios, R., Hernández-López, A., García-Crespo, A., & Soto-Acosta, P. (2010). A Study of Emotions in Requirements Engineering. Organizational, business, and technological aspects of the knowledge society. *Communications in Computer and Information Science*, 112, 1-7. doi: 10.1007/978-3-642-16324-1_1 (Scopus Source Normalized Impact per Paper, SNIP 2010, 0.110)
- Hernández-López, A., Colomo-Palacios, R., García-Crespo, Á., Martín-Paniagua, F., & Soto-Acosta, P. (2010). Personnel Performance Management in IT eSourcing Environments. In N. S. Shi & G. Silvius (Eds.), *Enterprise IT Governance, Business Value and Performance Measurement* (pp. 202-219): IGI-Global. doi: 10.4018/978-1-60566-346-3.ch013
- Hernández-López, A., Colomo-Palacios, R., García-Crespo, Á., & Soto-Acosta, P. (2010). Team Software Process in GSD Teams: A study of new work practices and models. *International Journal of Human Capital and Information Technology Professionals*, 1(3), 32-53. doi: 10.4018/jhcityp.2010070103
- Hernández-López, A., Colomo-Palacios, R., García-Crespo, Á., & Soto-Acosta, P. (2010). Trust Building Process for Global Software Development Teams. A review from the Literature. *International Journal of Knowledge Society Research*, 1(1), 65-82. doi: 10.4018/jksr.2010010105

2009

- Hernández-López, A., Colomo-Palacios, R., & García-Crespo, Á. (2009). Recomendaciones para la adopción de prácticas de gestión del capital humano en entornos de outsourcing. Integración de eSCM-CL con People-CMM. *Revista Española de Innovación y Calidad e Ingeniería del Software*, 5(1), 20-37.

Anexo G - Diagramas de procesos de proyectos de desarrollo software en las organizaciones participantes

Caso 3

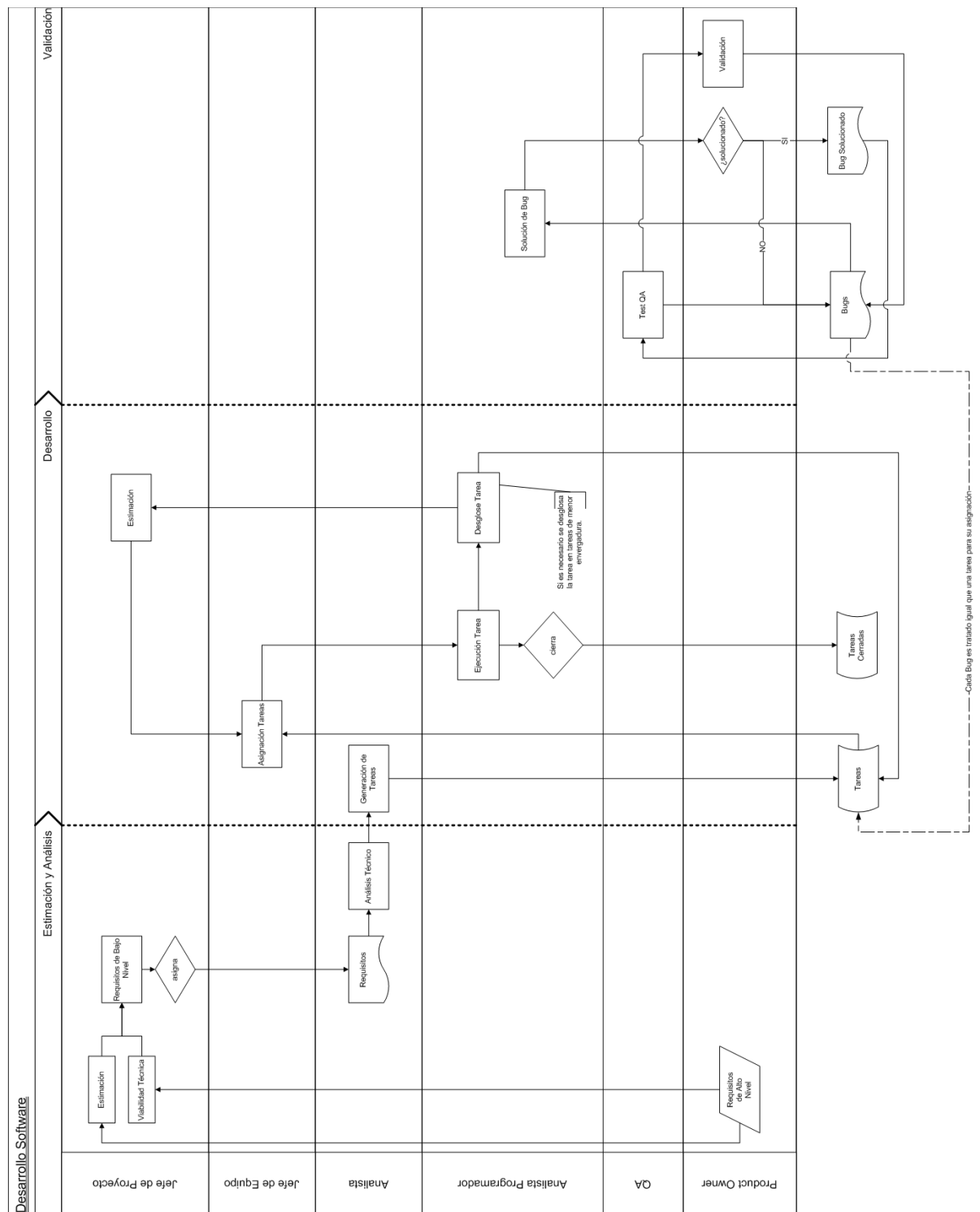


Ilustración 28. Proceso de trabajo en los proyectos de desarrollo y mantenimiento del Caso

Caso 4

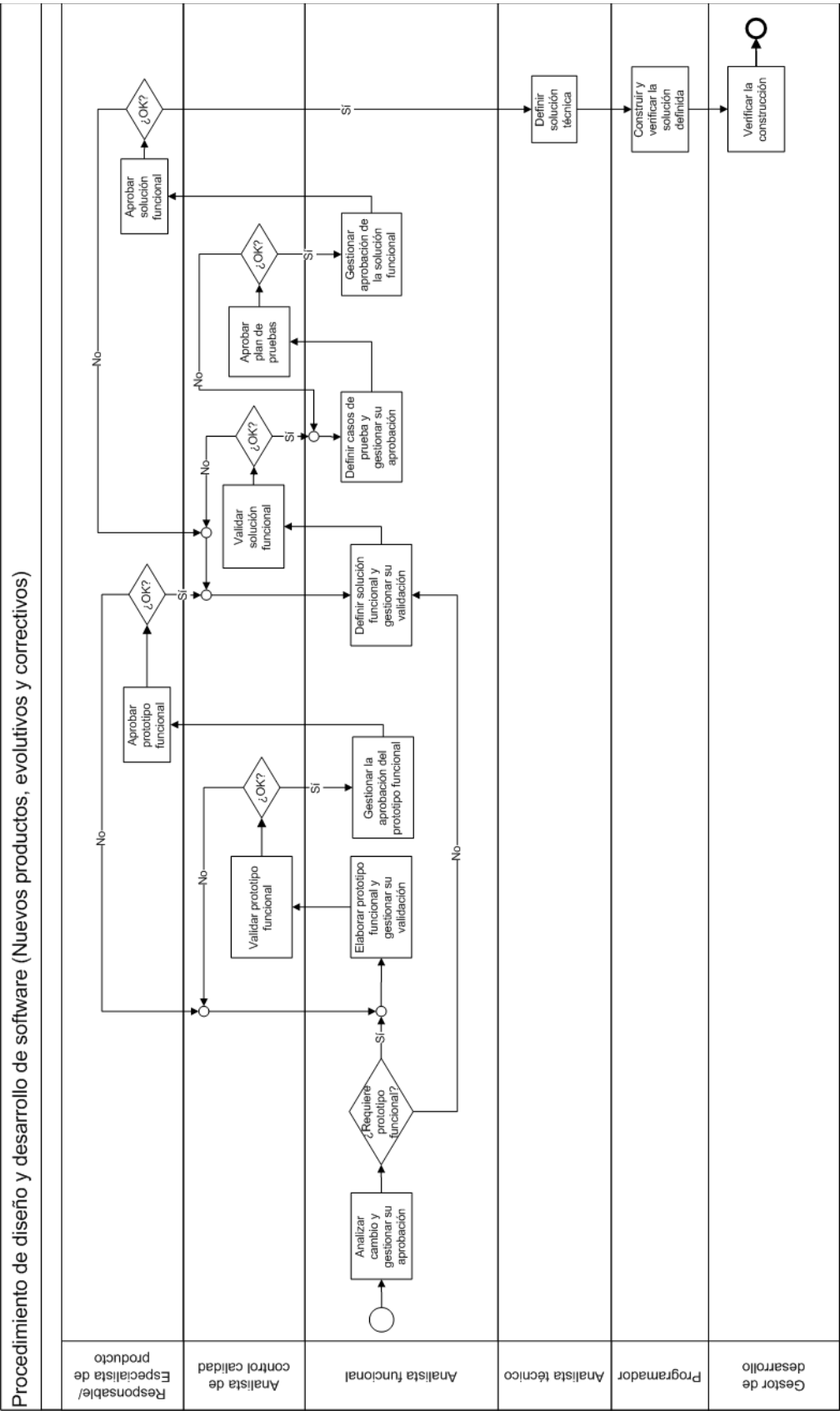


Ilustración 29. Proceso de trabajo en los proyectos de desarrollo y mantenimiento del Caso

Anexo H - Cuestionarios utilizados en los casos de estudio

Motivación intrínseca

Este cuestionario fue creado por Warr, Cook y Wall (1979).El resultado es la suma de todos los valores ya que no hay ítems con escala inversa.

	Por favor utilice la escala siguiente para contestar a cada pregunta, redondeando el número apropiado.	No, estoy totalmente en desacuerdo	No, estoy muy en desacuerdo	No, estoy un poco en desacuerdo	No estoy seguro/a	Sí, estoy un poco de acuerdo	Sí, estoy muy de acuerdo	Sí, estoy totalmente de acuerdo
1	Siento una sensación de satisfacción personal cuando hago mi trabajo bien.	1	2	3	4	5	6	7
2	Mi opinión de mí mismo disminuye cuando hago mi trabajo mal.	1	2	3	4	5	6	7
3	Me enorgullezco de hacer mi trabajo lo mejor que puedo.	1	2	3	4	5	6	7
4	Me siento triste cuando mi	1	2	3	4	5	6	7

	Por favor utilice la escala siguiente para contestar a cada pregunta, redondeando el número apropiado.	No, estoy totalmente en desacuerdo	No, estoy muy en desacuerdo	No, estoy un poco en desacuerdo	No estoy seguro/a	Sí, estoy un poco de acuerdo	Sí, estoy muy de acuerdo	Sí, estoy totalmente de acuerdo
	trabajo no está en mi nivel habitual							
5	Me gusta mirar hacia atrás en el trabajo del día y tener una sensación de trabajo bien hecho.	1	2	3	4	5	6	7
6	Trato de pensar en maneras de hacer mi trabajo de manera eficaz.	1	2	3	4	5	6	7

Satisfacción

Se seleccionaron algunas de las dimensiones del cuestionario desarrollado por Spector (1985)²⁹:

- Naturaleza del trabajo = $\text{Inv}(8)+17+27+35$
- Condiciones de trabajo = $\text{Inv}(6)+15+\text{Inv}(24)+\text{Inv}(31)$
- Comunicación = $9+\text{Inv}(18)+\text{Inv}(26)+\text{Inv}(36)$

²⁹ Las traducciones de las preguntas han sido obtenidas de la tesis doctoral Conrado Marion-Landais (1993). A Cross Cultural Study of Leader-Member Exchange Quality and Job Satisfaction as Correlates of Intradynamic Work-Value Similarity. University of South Florida. En esta traducción, el ítem 35 aparece como el 33, y el 36 aparece como 34. Se ha realizado una comprobación ítem a ítem con respecto al original a partir de dicha traducción para contrastarlo y adecuarlo a la numeración original.

donde $Inv(X) \Rightarrow 7\text{-Valor}$

Se han dejado fuera las siguientes dimensiones: compensación, promoción, supervisión, beneficios complementarios, recompensas contingentes, y compañeros.

	Por favor utilice la escala siguiente para contestar a cada pregunta, redondeando el número apropiado.	Totalmente en desacuerdo	Muy en desacuerdo	En desacuerdo	De acuerdo	Muy en acuerdo	Totalmente de acuerdo
6	Muchas de nuestras reglas y procedimientos dificultan el hacer un buen trabajo.	1	2	3	4	5	6
8	A veces siento que mi trabajo no tiene sentido.	1	2	3	4	5	6
9	La comunicación aparenta ser buena en esta compañía.	1	2	3	4	5	6
15	Mis esfuerzos para hacer un buen trabajo raramente son bloqueados por la burocracia.	1	2	3	4	5	6
17	Me gusta hacer las cosas que hago en mi trabajo.	1	2	3	4	5	6
18	Las metas de esta empresa no me son claras.	1	2	3	4	5	6
24	Tengo demasiado que hacer en el trabajo.	1	2	3	4	5	6
26	A menudo siento que no sé lo que está pasando con la compañía.	1	2	3	4	5	6
27	Siento orgullo en hacer mi trabajo.	1	2	3	4	5	6
31	Tengo demasiado papeleo.	1	2	3	4	5	6
35	Mi trabajo es agradable.	1	2	3	4	5	6

	Por favor utilice la escala siguiente para contestar a cada pregunta, redondeando el número apropiado.	Totalmente en desacuerdo	Muy en desacuerdo	En desacuerdo	De acuerdo	Muy en acuerdo	Totalmente de acuerdo
36	Las tareas asignadas no siempre son totalmente explicadas.	1	2	3	4	5	6

